NEW AGE

COMPUTER NETWORK

S.S. Shinde

NEW AGE INTERNATIONAL PUBLISHER

COMPUTER NETWORK

This page intentionally left blank

COMPUTER NETWORK

S.S. Shinde
M.E. (Electronics)
Lecturer
Annasaheb Dange College of
Engineering & Technology
Ashta (Sangli), Maharashtra

PUBLISHING FOR ONE WORLD

NEW AGE INTERNATIONAL (P) LIMITED, PUBLISHERS
New Delhi • Bangalore • Chennai • Cochin • Guwahati • Hyderabad
Jalandhar • Kolkata • Lucknow • Mumbai • Ranchi
Visit us at **www.newagepublishers.com**

Published by New Age International (P) Ltd., Publishers

ISBN (13) : 978-81-224-2852-0

PUBLISHING FOR ONE WORLD

NEW AGE INTERNATIONAL (P) LIMITED, PUBLISHERS
4835/24, Ansari Road, Daryaganj, New Delhi - 110002
Visit us at **www.newagepublishers.com**

ACKNOWLEDGEMENT

It gives me a great pleasure in bringing out this book on **COMPUTER NETWORK** which covers most of the concepts from computer and communication network.

I would like to warmly thank my parents, my husband Mr. Sunil Shinde for providing advice, encouragement and support at various stages of the preparation of this book. I also thank them for reminding us that there are things in life beyond writing books.

I am very much thankful to Principal and Management of Annasaheb Dange College of Engineering and Technology, Ashta, for their encouragement and moral support. Finally I thank to all staff of New Age Intertions, Publishes for essential support throughout the arduous process of getting the book into print.

I welcome any constructive criticism of the book and will be grateful for any appraisal by the readers.

SUNITA S. SHINDE

This page intentionally left blank

PREFACE

Unlike the other computer networking books, this book is organized in a bottom-up manner, i.e., it begins at the physical layer and works its way down towards the application layer. The bottom-up approach has several important benefits.

1. It places emphasis on the physical layer, which has been the high"growth area" of hardware and computer networking and communication. Also at top layer, indeed, many of the recent revolutions in computer networking. In preparing the first edition of this book, we believed that the application layer would remain the highest growth area in the field, both in terms of research and actual deployment.
2. Our experience as instructor has been that teaching networking applications near the beginning of the course is a powerful motivational tool. Students are thrilled to learn about how networking applications work- application such as e-mail and the Web which most students use on a daily basis.
3. The bottom-up approach enables instructors to introduce network hardware development at an early stage. Students not only see how popular applications and protocols work, but also learn how easy it is to create their own network applications and application-level protocols.

Thus, with the bottom-up approach, students get early exposure to the notions of hardware programming interfaces(APIs), service models, protocols, and other important concepts.

The chapter by chapter description of the book is as follows:

Chapter 1: This chapter describes basics of communication networks.

Chapter 2: This chapter describes modulation techniques and concepts of noise.

Chapter 3: This chapter describes about multiplexing techniques.

Chapter 4: This chapter explains introduction to computer networks.

Chapter 5: This chapter focuses on different network concepts and components.

Chapter 6: This chapter describes the physical layer structure and it covers theoretical basis for data communication, about transmission media, RS-232, and about modem.

Chapter 7: This chapter describes the data link layer structure, which covers basics of data link layer, error control, detection and correction, data link layer protocols and HDLC.

Chapter 8: This chapter describes the medium access sub layer, which covers channel allocation, medium access control layer protocols, IEEE standards, satellite networks.

Chapter 9: This chapter explains about the network layer as routing algorithms and congestion control.

Chapter 10: This chapter describes internetworking concepts, TCP/IP protocol suit and internet control protocols.

Chapter 11: This chapter describes about transport layer.

Chapter 11: This chapter focuses on presentation and application layer.

The book contains questionnaires.

AUTHOR

CONTENTS

Preface (*vii*)

1. BASIC WORKING CONCEPT AND SYSTEM 1

1.1 Concept of Communication Network 1

1.2 Types of Communication Network 3

1.3 Channels and Circuits 4

1.4 Signals and Transmission 7

1.5 Channel Speed and Bit Rate 12

1.6 Online and Offline Systems 13

1.7 Interactive and Non-interactive Systems 14

Questionnaires 15

2. COMMUNICATION SYSTEM AND NOISE 16

2.1 Concept of Modulation 16

2.2 Amplitude Modulation 17

2.3 AM Bandwidth Requirement 18

2.4 Frequency Modulation 21

2.5 FM Bandwidth Requirement 22

2.6 Concept of Noise 30

2.7 Noise Figure and Noise Temperature 33

Questionnaires 35

3. MULTIPLEXING 36

3.1 Concept of Multiplexing 36

3.2 Frequency Division Multiplexing 37

3.3 Time Division Multiplexing 40

3.4 Wavelength Division Multiplexing 43

Questionnaires 44

4. INTRODUCTION TO COMPUTER NETWORK 45

4.1 Need of Computer Network 46
4.2 Advantages of Computer Network 46
4.3 Uses of Computer Network 47
4.4 Network Models 48
4.5 Categories of Networks and Internetworks 48
4.6 Line Configurations 51
4.7 Topology 52
4.8 Study of Reference Models 57
4.8.1 Need of Layers 57
4.8.2 Design Issues of Layers 59
4.8.3 ISO/OSI Model 60
4.8.4 TCP/IP Model 63
4.8.5 A Comparison of OSI and TCP Reference Model 66
4.8.6 Asynchronous Transfer Mode (ATM) 67
4.9 Network Examples 68
Questionnaires 76

5. NETWORK CONCEPTS AND COMPONENTS 77

5.1 Network Concepts 77
5.1.1 Wireless Networks 77
5.1.2 Layered Approach 78
5.1.3 Interfaces 79
5.1.4 Services 80
5.1.5 Protocols 81
5.1.6 Brief Study of X.25 Protocol 83
5.1.7 Intranet and Extranet 90
5.2 Network Components 91
5.2.1 Cabling and Connector Standards 91
5.2.2 Network Interface Card 99
5.2.3 Bridges/Switches 100
5.2.4 Routers 104
5.2.5 Concentrators 105
5.2.6 Hubs 106
5.2.7 Repeaters 107

5.2.8 Gateways 109
5.2.9 ISDN 110
Questionnaires 122

6. PHYSICAL LAYER 124

6.1 Physical Layer Characteristics 124
6.2 The Theoretical Basis for Data Communication 125
6.2.1 Fourier Analysis 125
6.2.2 Bandwidth Limited Signals 128
6.2.3 The Maximum Data Rate of Channels 131
6.3 Transmission Media 132
6.3.1 Guided Media 132
6.3.2 Unguided Media 141
6.4 Transmission Impairment 150
6.5 Design Issues of Physical Layer 152
6.6 EIA-232-D Digital Interface 155
6.7 EIA-232-D Interface Standard 157
6.8 Modems 161
6.8.1 Introduction 161
6.8.2 Types of Modems 162
6.8.3 Block Schematic of Modems 166
Questionnaires 169

7. DATA LINK LAYER 170

7.1 Data Link Layer Design Issues 170
7.2 Services Provided to the Network 170
7.3 Framing Methods 173
7.4 Error Control-Detection and Correction 175
7.5 Flow Control 186
7.6 Elementary Data Link Protocols 186
7.7 Sliding Window Protocols 196
7.7.1 Stop and Wait Sliding Window Protocol 198
7.7.2 Sliding Window Protocol with Go-Back-N 200
7.7.3 Sliding Window Protocol with Selective Repeat 201
7.8 High Level Data Link Control (HDLC) 203

7.8.1 Introduction 203
7.8.2 Types of Stations 204
7.8.3 Modes of Operation 207
7.8.4 HDLC Frame Formats 207
7.8.5 Additional Features 209
7.9 Data Link Layer In the Internet 210
7.9.1 SLIP 211
7.9.2 PPP 212
Questionnaires 213

8. MEDIUM ACCESS SUBLAYER 215
8.1 Channel Allocation 215
8.1.1 Static Channels Allocation 215
8.1.2 Dynamic Channel Allocation 216
8.2 Types of Access Protocol 217
8.2.1 ALOHA 217
8.2.2 CSMA(Carrier Sense Multiple Access) 221
8.2.3 WDMA(Wavelength Division Multiple Access) 225
8.3 IEEE Standards 227
8.3.1 IEEE 802.3: Ethernet 227
8.3.2 IEEE 802.4: Token Bus 229
8.3.3 IEEE 802.5: Token Ring 230
8.4 High Speed LANs 231
8.4.1 FDDI (Fiber Distributed Data Interface) 231
8.4.2 Fast Ethernet 235
8.5 Satellite Networks 237
8.5.1 Polling 237
8.5.2 ALOHA 238
8.5.3 FDM(Frequency Division Multiplexing) 239
8.5.4 TDM(Time Division Multiplexing) 239
8.5.5 CDMA(Code Division Multiple Access) 241
Questionnaires 242

9. NETWORK LAYER 243
9.1 Design Issues 243
9.2 Routing Algorithms 248

9.2.1 Shortest Path Routing 249
9.2.2 Flooding 250
9.2.3 Flow Based Routing 251
9.2.4 Distance Vector Routing 253
9.2.5 Link State Routing 255
9.2.6 Broadcast Routing (Routing for Mobile Hosts) 257
9.3 Congestion Control 257
9.3.1 Congestion Control Principles 258
9.3.2 Congestion Prevention Policies 259
9.3.3 Leaky Bucket Algorithms 260
9.3.4 Token Bucket Algorithm 262
9.3.5 Choke Packets 263
9.3.6 Load Shedding 266
9.3.7 Jitter Control 267
Questionnaires 267

10. INTERNETWORKING 268

10.1 Internetworking Devices 268
10.2 Concatenated Virtual Circuits 269
10.3 Connectionless Internetworking 270
10.4 Tunneling 271
10.5 Routing 272
10.6 Fragmentation 273
10.7 Firewalls 275
10.8 TCP/IP Protocol Suit Overview 277
10.8.1 TCP/IP versus OSI Model 278
10.8.2 Network Layer In the Internet 278
10.8.3 Internet Protocol 279
10.8.4 IP Addresses 281
10.8.5 Subnetting 283
10.8.6 Internet Control Protocol 284
Questionnaires 289

11. TRANSPORT LAYER 290

11.1 Service Primitives 290

11.2 Elements of Transport Protocol 293

11.3 TCP(Transmission Control Protocol) 298

11.4 UDP(User Datagram Protocol) 304

Questionnaires 305

12. UPPER LAYER 306

12.1 Network Security 306

12.2 Cryptography 308

12.3 Secret Algorithms 310

12.4 Encryption with Public Keys, Private Keys 312

12.5 Session Layer Operation and Services 314

12.6 ASN.1-Abstract Syntax Notation 1 315

12.7 SNMP(Simple Network Management Protocol) 317

12.8 DNS(Domain Name System) 318

12.9 Application Layer Services 321

12.10 Study of Internet Tools 322

12.10.1 Email(Electronic Mail) 322

12.10.2 USENET News 324

12.10.3 WWW(World Wide Web) 326

12.10.4 HTTP(Hyper Text Transfer Protocol) 327

12.10.5 HTML(Hyper Text Markup Language) 330

Questionnaires 331

Appendices

Appendix A 333

Appendix B 333

Appendix C 352

Appendix D 358

Appendix E 359

Appendix F 362

Appendix G 365

Glossary 387

Index 393

Chapter 1

BASIC WORKING CONCEPT AND SYSTEM

INTRODUCTION

Welcome to the world of 'Information Technology' if you aren't already there, that is. In this chapter you will go through the routine procedure of learning about the basic working concept of communications, and also various communication systems.

1.1 CONCEPT OF COMMUNICATION NETWORK

Communication

The process of sharing resources, exchanging data, providing back up for each other and allowing employees and individuals to perform their work from any location.

Some of the common **objectives** of the computer communication network are :

- To provide sharing of resources such as information or processors.
- To provide inter-process communication among users and processors.
- To provide distribution of processing functions.
- To provide centralised control for a geographically distributed system.
- To provide centralised management and allocation of network resources : host processors, transmission facilities.
- To provide compatibility of dissimilar equipment and software.
- To provide network users with maximum performance at minimum cost.
- To provide an efficient means of transport large volumes of data among remote locations.

Structure of the Communication System

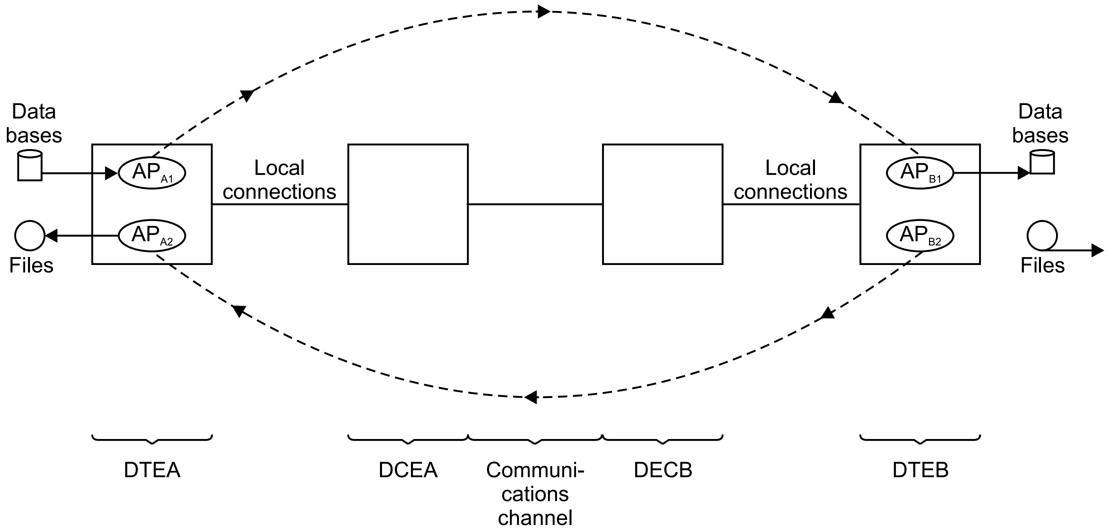


Fig. 1.1 Communication system

Figure 1.1 illustrates a simple data communications system. The **application process** (AP) is the end-user application. It usually consists of software such as a computer program. Typical examples are an accounts receivable program, a payroll program, and airline reservation system, an inventory control package or a personnel system.

In figure 1.1, site A could execute an application process (AP_{A1}) in the form of a software program to access an application process at site B (which is, in this case, a program (AP_{B1}) and a database). Figure 1.1 also shows a site B program (AP_{B2}) accessing a file at site A through an application program (AP_{A2}).

[In this book application process term is used to describe end-user applications, unless otherwise noted].

The application resides in the data terminal equipment, or DTE. DTE is a generic term used to describe the end-user machine, which is usually a computer or terminal. The DTE could be a large mainframe computer, such as a large IBM or ICL machine, or it could be a smaller machine, such as a terminal or a personal computer. The DTE takes many forms in the industry. Here are several **examples:**

- a workstation for an air traffic controller.
- an automated teller machine in a bank.
- a point-of-sale terminal in a department store.
- a sampling device to measure the quality of air.
- a computer used to automate the manufacturing process in a factory.
- an electronic mail computer or terminal.
- a personal computer in the home or office.

The **function** of communications network is to interconnect DTEs so they can share resources, exchange data, provide back up for each other, and allow employees and individuals to perform their work from any location.

Figure 1.1 shows that a network provides **logical** and **physical** communications for the computers and terminals to be connected. The applications and files use the physical channel to effect logical communications. Logical, in this context, means the DTEs are not concerned with the physical aspects of the communications process. Application A_1 need only issue a logical Read request with an identification of the data. In turn, the communications system is responsible for sending the Read request across the physical channels to application B1.

Figure 1.1 also shows the **data circuit-terminating equipment**, or DCE (also called **data communications equipment**). Its function is to connect the DTEs into the communication line or channel. The DCEs designed in the 1960s and 1970s were strictly communications devices. However, in the last decade the DCEs have incorporated more user functions, and today some DCEs contain a portion of an application process. Notwithstanding, the primary function of the DCE remains to provide an interface of the DTE into the communications network. The familiar modem is an example of a DCE.

The interfaces are specified and established through protocols. Protocols are agreements on how communications components and DTS are to communicate with each other. They may include actual regulations which stipulate a required or recommended convention or technique. Typically, several levels of interface and protocols are required to support an end-user application.

1.2 TYPES OF COMMUNICATION

There are basically three different types of communication.

1. Processor to Processor

Processor to processor communication means the communication that takes place between two or more computers when working in tandem for such things as timing and synchronization.

This normally refers to communication between two or more computers to interchange large quantities of data, such as the bulk update of files, records, and so on.

This communication tends to be very fast and often takes place over a distance of a few feet between computers in the same room. As the capacity of telecommunications networks increases, computers working in tandem or parallel can be dispersed over a wide geographic area.

2. Personal Computer or Dumb Terminal to Host Computer

The personal computer can send, receive, and store information from another larger computer, usually referred to as the host computer. Many personal computers or dumb terminals can access the same host computer. The host computer can be located in the next room or many thousands of miles away.

The difference between a personal computer or a dumb terminal in this case is that the personal computer has its own local processing and storage, whereas the terminal is a simple, screen-based input/output device.

3. Personal Computer to Personal Computer

Personal computers can communicate with each other on a one-to-one basis.

They exchange information freely with one another, as opposed to communicating with a single host computer. Communication between personal computers can be at a high speed over short distances via a local area network (LAN) or at slower speeds or over any distance using the conventional telephone network.

1.3 CHANNELS AND CIRCUITS

The generic interconnection between a message source and its destination, or message sink, is called a channel.

Two types of channel configurations are there :

1. Point-to-Point Channel

In this network two DTE devices are connected by telephone lines.

Here the message is send from one DTE to another DTE, is received by all intermediate DTEs, and stored their until the request output line is free.

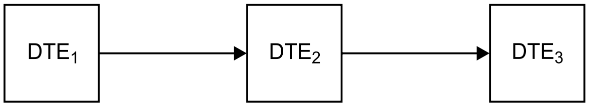


Fig. 1.2 Point-to-point channel

2. Broad-Casting Channels/Multidrop Channels

In this configuration more than two DTE devices are connected to the same channel just like a transmission line.

Here block of information is sent by sender (DTE) which is in the form of packets. Then each DTE of the network checks address of each packet and accepts only those that are addressed to it.

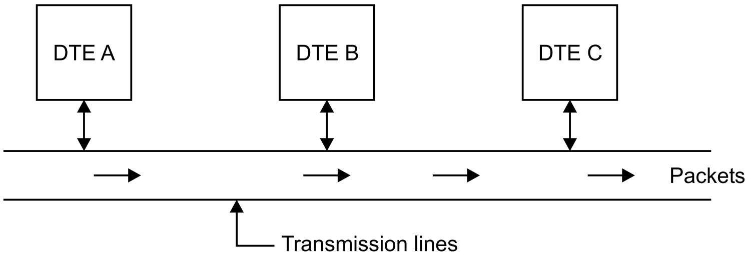


Fig. 1.3 Multidrop channels

Most of us will first think of a channel as the connection between a particular broadcasting station and our television receivers, with different channels allocated to different stations. An associated concept is that of a circuit, which is the complete path connecting a source with its destination(s).

In practice, “circuit” and “channel” are sometimes used interchangeably.

There are three modes of circuit use in communications.

1. Simplex

A simplex arrangement allows communication in one direction only. Here the role of source and destination are permanently assigned.

It is common in television and commercial radio.

It is not as common in data communication because of the one-way nature of the process. Simplex system are found in some applications such as telemetry, burglar alarm.

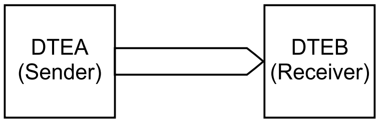


Fig. 1.4 Simplex

2. Half duplex

Half duplex arrangement allows communication in both directions but only one direction at a time. Here the roles of source and destination are allowed to change.

This is also called a Two-Way-Alternate i.e., (TWA).

Half-duplex systems found in many systems, such as inquiry/response applications, wherein a DTE sends a query to another DTE and waits for the applications process to access and/or compute the answer and transmit the response back.

Terminal-based systems (keyboard terminals and terminals with CRT screens) often use half-duplex techniques.

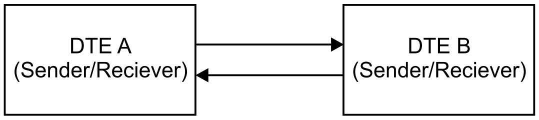


Fig. 1.5 Half duplex

3. Full duplex (or Duplex)

Full duplex arrangement allows communication in both directions simultaneously.

This is also called Two-Way-Simultaneous (TWS).

Full duplex or (duplex) provides for simultaneous two-way transmission, without the intervening stop-and-wait aspect of half-duplex.

Full duplex is widely used in applications requiring continuous channel usage, high throughput, and fast response time.

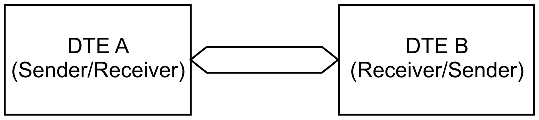


Fig. 1.6 Full duplex

Thus far, the terms half-duplex and full-duplex (duplex) have been used to describe how data move across the circuit. We have focused on these terms as they are used in the data communication industry, which is described in Figure 1.7.

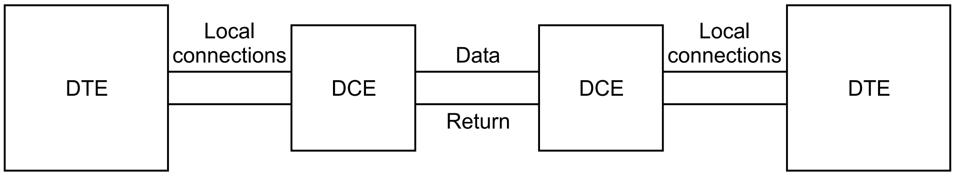


Fig. 1.7(a) Two-wire-circuit

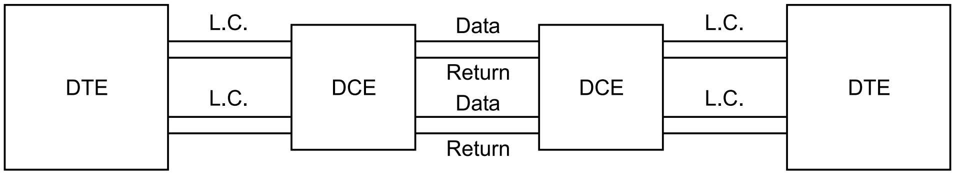


Fig. 1.7(b) Four-wire-circuit

In telephone communications, the term **two-wire** and **four-wire circuits** are used to describe the channel. One wire is for the transmission of data and the other is for the return circuit. In a four-wire circuit, two pairs of two wires exist – two for the data and two for the return circuits. The telephone company usually configures a two-wire circuit as a switched dial-up circuit and a four-wire circuit as a leased, nonswitched circuit. However, exceptions exit and the reader is encouraged to check with the specific telephone company.

The advantage cited earlier concerning communications networks cannot be realized without the addition of an important component to the system.

This component is the data switching equipment, or DSE. Figure 1.8. illustrate the use of the DSE in conduction with the DTE and DCE. As the name implies, the major function of the DSE is to switch or route traffic (user data) through the network to the final destination. The DSE provides the vital functions of network routing around failed or busy devices and channels. The DSE may also route the data to the final destination through intermediate components perhaps other switches.

Figure 1.8. illustrates a simple arrangement of the DCE, DTE and DSE in the network.

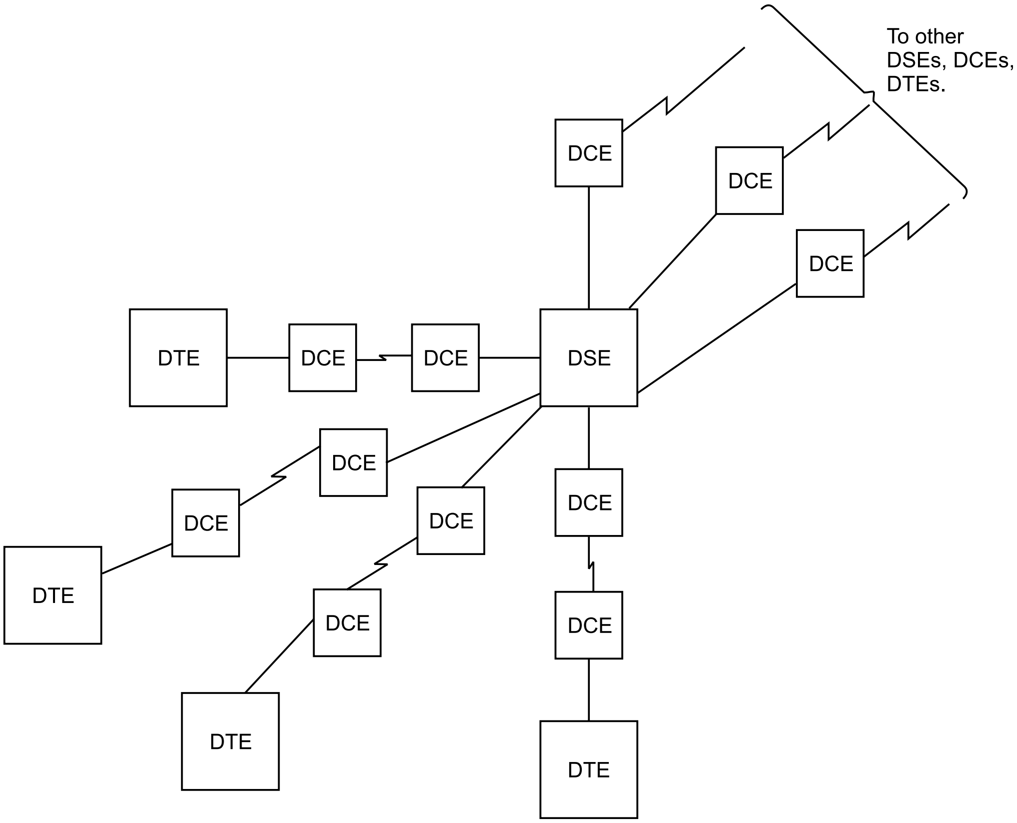


Fig. 1.8 Data switching equipment (DSE)

1.4 SIGNALS AND TRANSMISSION

SIGNALS

Signal is referred to as a set of information or data.

Analog Signals

Voice communication generate acoustical waveforms which propagate through the air, passing with analog signals. In effect voice communications are physical energy.

When one speaks, oscillating waveforms of high/low air pressures are created. These waveforms are the analog waveforms and signals or set of data passes through these waveforms are called analog signal.

Analog signals are continuous, repeating occurances and they are nondiscrete.

Nondiscrete—gradually changing from high to low pressure.

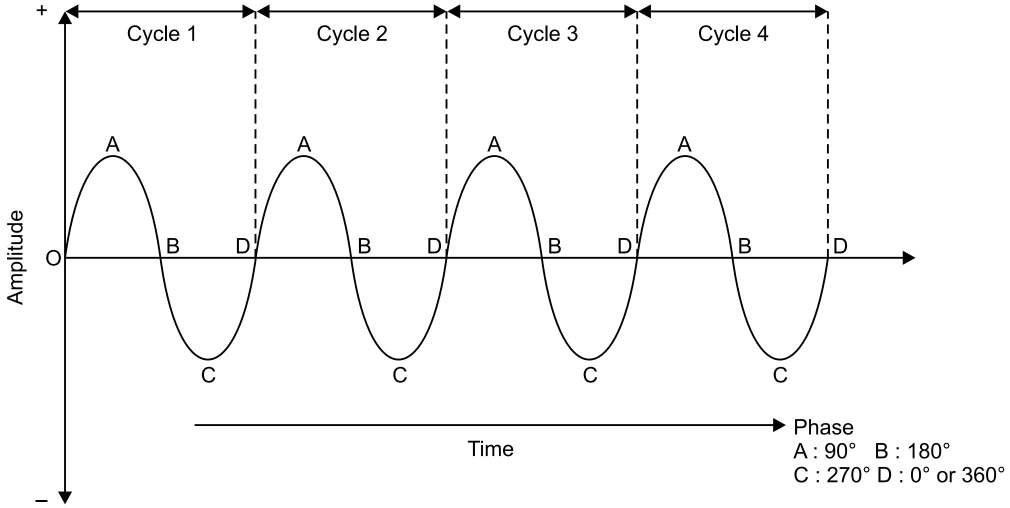


Fig. 1.9 The Analog signal

Analog signals are distinguished by an infinite possibility of values to represent the infinite possible variations of some transmission characteristic.

Analog signals has three major characteristics.

- Amplitude
- Frequency
- Phase

Amplitude of the signal is a measurement in relation to its voltages, which can be zero, plus or minus value. It gradually increases in positive value, then traverse the zero voltage to the negative voltages, and return to zero again.

Frequency describes the No. of complete cycles per second.

Phase of the signal represents the point, the signal has reached in the cycle.

Digital Signal

When DTEs communicate with each other by use of a telephone path. However, DTEs “talk” in digital forms.

Digital signals are defined by a limited No. of representative value.

As shown in figure 1.10 the digital waveform looks considerably different from the analog waveform and the data passes through the digital waveform is called as digital signal.

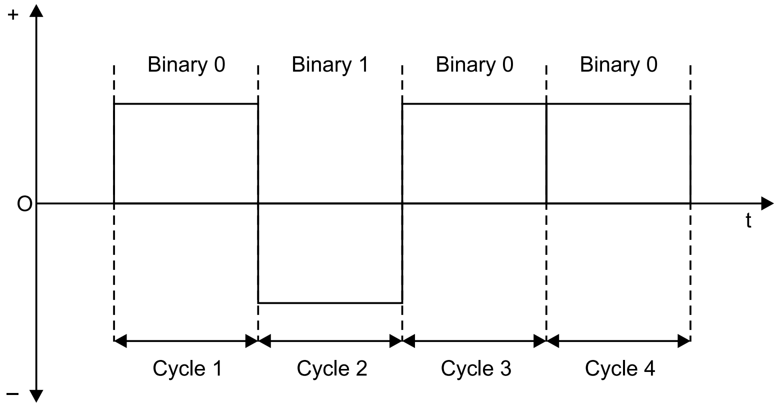


Fig. 1.10 The digital signal

It is continuous, repeats itself and periodic but it is discrete.

Discrete–It has very abrupt changes in its voltage state.

Digital signals have two states voltages which is 0 volts represents binary 0 or lower amplitude and 5 or + volts represents binary 1 or higher amplitude.

Due to the electrical properties of the channel, the signal is actually less discrete and square shaped.

TRANSMISSION

There is always a need to exchange data, commands and other control information between a computer and its terminals or between two computers. This information is in the form of bits.

Transmission refers to movement of bits, data or signals over some physical medium connecting two or more digital devices.

Bits or data can be transmitted in two ways namely, parallel transmission or serial transmission.

Parallel transmission

In parallel transmission, all the bits of a byte are transmitted simultaneously on separate wires as shown in figure 1.11 and multiple circuits interconnecting the two devices are, therefore, required.

It is practical only if two devices, e.g., a computer and its associated printer are close to each other.

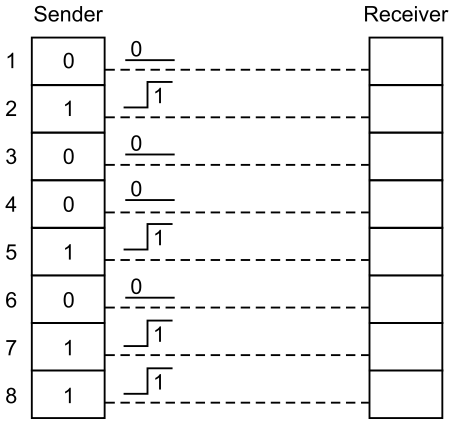


Fig. 1.11 Parallel transmission

Serial transmission

In serial transmission, bits are transmitted serially one after the other as shown in figure 1.12. The least significant bit (LSB) is usually transmitted first.

Note that as compared to parallel transmission, serial transmission require only one circuit interconnecting the two devices. Therefore, serial transmission is suitable for transmission over long distances.

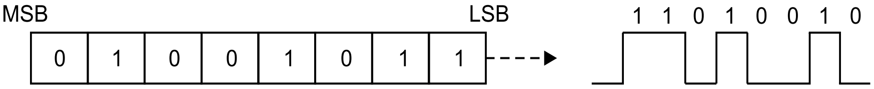


Fig. 1.12 Serial transmission

One major difficulty in data transmission is that of synchronising the receiver with the sender. Clearly, this is particularly true in serial data transfer, where the receiver must be able to detect the beginning of each new character in the bit stream being presented to it. Obviously, if it is unable to achieve this, it will not be able to interpret the incoming bit stream correctly.

There are three major **modes** of transmission.

- Asynchronous transmission.
- Synchronous transmission.
- Isochronous transmission.

Asynchronous transmission

Asynchronous transmission sends only 1 character at a time where a character is either a letter of the alphabet or number or control character, which is shown in fig. 1.13.

Preceding each character is a start bit and ending each character is 1 or more stop bits.

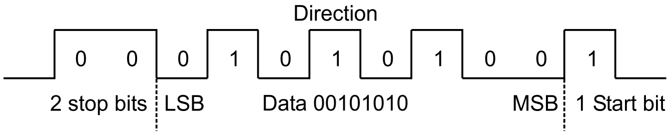


Fig. 1.13 Asynchronous transmission

Asynchronous transmission is simple and inexpensive to implement. It is used mainly with serial ports and dial up connections. It requires start and stop bit for each character, which adds a high overhead to transmission.

Advantages

- In this approach each individual character is complete in itself – therefore, if a character is corrupted during transmission. Its successor and predecessor will be unaffected.
- It is particularly suited for applications where the characters are generated at irregular intervals.

  e.g., data entry from the keyboard.

Disadvantages

- Successful transmission inevitably depend on the recognition of the start bits – clearly these can be easily missed, or occasionally spurious start bits can be generated by line interference.
- A high proportion of the transmitted bits are uniquely for control purposes and thus carry no useful information.
- As a result of the effects of distortion the speed of transmission is limited. Thus, in conclusion, asynchronous serial transmission is normally only used for speeds of up to 3000 bits per second, with only simple, single-character error detection.

Synchronous transmission

When higher data rates are desirable, synchronous transmission is preferred to transfer a large number of characters consecutively.

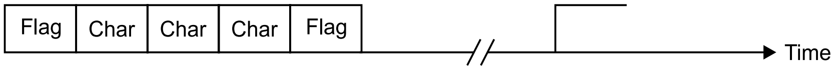


Fig. 1.14 Synchronous serial transmission

Synchronous transmission is more efficient as little as only 4 bytes are required to transmit up to k bits. Synchronous transmission is more difficult and expensive to implement. It is used with all higher communication transfer rates : Ethernet, Token Ring. Historically, synchronous communications were operating over 2400/4800 baud modems on point-to-point communications.

Advantages

- The amount of central information that requires to be transmitted is restricted to only a few characters at the start of each block.
- The system is not so prone to distortion as asynchronous communication and can thus be used at higher speeds.

Disadvantages

- If an error does occur the rather than just a single character the whole block of data is lost.
- The sender cannot transmit characters simply as they occur and consequently has to store them. Until it has built up a block – thus the system is unsuitable for applications where characters are generated at irregular intervals.

Isochronous transmission

The third mode of transmission, called isochronous, which is shown in figure 1.15, is sometimes used for devices that have special requirements for a periodic scan or rotation rate, as in the video data required to "paint" a picture on a television screen.

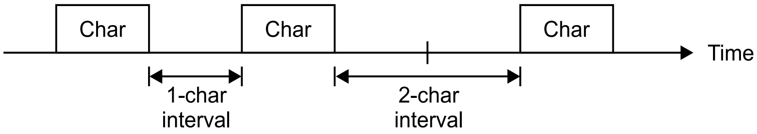


Fig. 1.15 Isochronous serial transmission

A start and stop bit are placed around each character as for asynchronous transmission but the time interval between characters must be exactly a multiple of the character length; precise clocking is required, just as for synchronous transmission.

This mode can also be used to allow asynchronous terminals to transmit data at speeds faster than normally possible with simple asynchronous transmission.

1.5 CHANNEL SPEED AND BIT RATE

The most elementary method a device uses to send a binary number on a communications path is to switch the signal on and off electrically, or to provide high or low voltages on the line to represent the 1s and 0s.

Regardless of how the data are represented on the path in the form of on/off states, levels of voltage, or directions of current flow. The communications channel is described by its capacity in the number of bits per second transmitted. Abbreviations for *bits per second* are "bits", "bps", "or bs". When one speaks of a 4800 bit/s line, it means a device sends 4800 bits per second through the channel. A bit is simply the representation of the electrical, optical, or electromagnetic state of the line: voltages, current, or some form of radio or optical signal. Seven or eight bits usually comprise a user-coded character, or byte.

A data communications channel utilizing conventional telephone lines is very slow. Below are some examples. For purposes of comparison, a channel is classified by categories of low speed, medium speed, and high speed.

Low speed: 0 – 600 bits per second.

Medium speed: 600 – 4800 bits per second.

High speed: 4800 – 9600 bits per second.

Only recently, in the last few years, has the industry successfully moved to 9.6 kilobits per second (kbit/s) on telephone channels. The typical speeds found beyond 9600 bits per second are 14,400, 19,200, 56,000, and 64,000 bit/s, and 1.544 megabit/s (1,544,000 bits per second) and 2.048 Mbit/s in Europe. The 1.544 megabits per second channel is the well-known T1 carrier. This offering is prevalent in transmissions such as high-speed digital channels and digitals switches.

The idea of a high-speed channel operating at 9.63 kbit/s is rapidly changing. With the proliferation of optical fiber tecnology, mbit speeds are becoming common place.

One might reasonably ask, why the slow speed? The answer is that DTEs and DCEs usually communicate through the telephone line. It was the most convenient and readily available path when the industry developed computers and began to interface them with terminals and other computers in the 1960s. The telephone channel is not designed for fast transmission between high-speed computers, but for voice transmission between people which does not require the speed associated with data transmission.

Moreover, the majority of DTEs remaining today are connected through the telephone line with an interface called EIA-232-D. This rather ancient interface is still the most pervasive way of connecting DTEs and DCEs to the telephone lines. The EIA-232-D is constrained to bit rates up to 20 kbit/s.

The transmission speed is characterized in two different ways.

- **Bit rate—**The number of binary digits transmitted per second.
- **Baud rate—**The number of discrete signalling elements transmitted per second.

Multiple bits per baud are also possible. In serial data transfer operations, bits are usually sent sequentionally, one bit carried in each signal time. Thus the bit rate and baud rate are equivalent for most serial connection. And in parallel data transfer operations, multiple bits are sent at each signal time, so the bit rate is equal to the baud rate multiplied by the No. of bits sent in parallel connection.

1.6 ONLINE AND OFFLINE SYSTEMS

Information can be collected and transmitted in two way, i.e., online, offline.

Online Systems

An online system may be defined as one in which the input data enter the computer directly, from the point of origination and/or output data are transmitted directly to where they are used.

Intermediate stages of writing data in magnetic tape, disc or offline printing are avoided.

A good example of an online system would be **a bank's automatic teller machine**. When a customer inserts his or her bank or credit card and makes a request to withdraw money the automatic teller machine establishes a connection with the computer which holds the account records of the customer and checks to see if there are sufficient funds in the account to cover the withdrawal. If there are, the machine hands over the money ; if there are not – no money.

Offline Systems

Offline systems may be defined as telecommunication data do not go directly into the computer but are written onto magnetic tape or disk for later processing.

In these system, data are prepared ready for transmission at a later date.

One example of offline system is an **electronic cash registers**.

Many electronic cash registers in shops are connected to a central computer, which records the amount of money taken and whose goods have been purchased. It is frequently not necessary for the central computer to get this information each time the cash register rings up a purchase. If a network of stores each had a cash register connected online to a central computer, there would be thousand of transactions passing to and fro every second. A more reasonable approach would be to receive a batch of figures from each cash register – say, at the end of each day. When the shop closed, the cash register would automatically send to the central computer details of the total amount of money taken and which goods have been sold computers that operate in this way are referred to as **batch processing systems** because they process data in batches sent at a predetermined time.

1.7 INTERACTIVE AND NON-INTERACTIVE SYSTEMS

There are another two types of communication systems.

Interactive System

Interactive means that the two devices communicating are exchanging information.

A system in which a person communicating with a computer in a conventions fashion is called an interactive system.

Interactive systems make different type of demands on the transmission and switching facilities to the traditional uses of telecommunications such as telephony, telegraph, broadcasting, etc. Prior to existance of new types of networks interactive system had to be built from existing telecommunications facilities.

Interactive system gives correct responses of data transmission. Most transmissions from human operators at terminals are interactive. Interactive systems have a high flow of data in both directions.

A good example of interactive system is **Facsimile machine**. When sending a document by facsimile, a connection must be made between the two devices. Once the connection is made and the document sent, the sending device cannot communicate any further. In some cases a low level of interaction may takes place, if the document received is full of errors, the receiving device may request that the document be transmitted but this is about the extent of the interaction.

Non-interactive System

A system which gives a very rudimentary response is the non-interactive systems. Here communication is almost entirely one way, i.e., from sender to the receiver.

In these systems, data travels in one direction only but control signals only travels in the other direction. Some online systems are also non-interactive systems.

An example of a non-interactive computer system could be a **store-and-forward message** or **electronic mail system**. Sending a message in this way from one terminal to another requires the message to pass first from the sender's terminal to another computer, where the message is stored. To get the message, the recipient established a connection with the computer storing the message and request that it be transferred to them. The person sending the message gets a response from the system to say, the message has been sent but nothing to say, that the person it was intended for has actually received it.

It is like putting a letter in the mail, if you put the letter in a post box, you know that the letter has been sent, but you know it arrived only if you get a reply.

QUESTIONNAIRES

1. Draw and explain the basic communication system.
2. Explain different types of communication in network system.
3. Explain different types of channels in communication.
4. Define circuit and give different types of circuits.
5. Explain different types of signals.
6. Explain various types of data transmission.
7. Describe on-line and off-line communication system.
8. Explain interactive and non-interactive communication system.
9. Write a note on channel speed and bit rate.

Chapter 2

COMMUNICATION SYSTEM AND NOISE

INTRODUCTION

Communication systems may be referred to as a collection of components that together accomplish a job or task. A communication system comprises all the components required to provide communication.

In this aspect the major emphasis is on the hardware, the software, and how the data are packaged. It is often difficult to establish the boundaries of a system; that is, deciding what is and what is not to be considered part of that system.

2.1 CONCEPT OF MODULATION

Modulation

Modulation techniques are methods used to encode digital information in an analog world. It is a process of modification of a signal to carry data. This signal is called the **carrier signal**. The data that modulates the carrier is **baseband signal**.

Suppose a square waves, as in digital data have a wide spectrum and thus are subject to strong attenuation and delay distortion. These effects make baseband (DC) signalling unsuitable except at slow speeds and over short distances.

To get around the problems associated with DC signalling, especially on telephone line, AC signalling is used. A continuous tone in the 1000–2000 Hz. range called a sine wave carrier is introduced. It's amplitude, frequency and phase can be modulated to transmit information.

There are three basic modulation techniques.

- AM (Amplitude Modulation)
- FM (Frequency Modulation)
- PM (Phase Modulation).

Need for Modulation

- In radio channel, try to send the (modulating) signal itself or else use an unmodulated carrier.
- An unmodulated carrier cannot be used to convey (carry) information.
- Here enough information about the instantaneous amplitude and frequency is transmitted to enable the receiver to recreate the original message.

2.2 AMPLITUDE MODULATION

Amplitude Modulation may be defined as information to be transmitted is superimposed on a carrier signal by varying the signal amplitude.

In this modulation two different voltage levels are used to represent 0 and 1 respectively. Here we could send several different amplitudes.

The amplitude of the carrier wave is varied in accordance with the signal to be sent. Amplitude modulated wave is made up of a number of sinusoidal components having a specific relation to one another.

Here the amplitude of a carrier signal is varied by the modulating signal, whose frequency is invariably lower than that of the carrier.

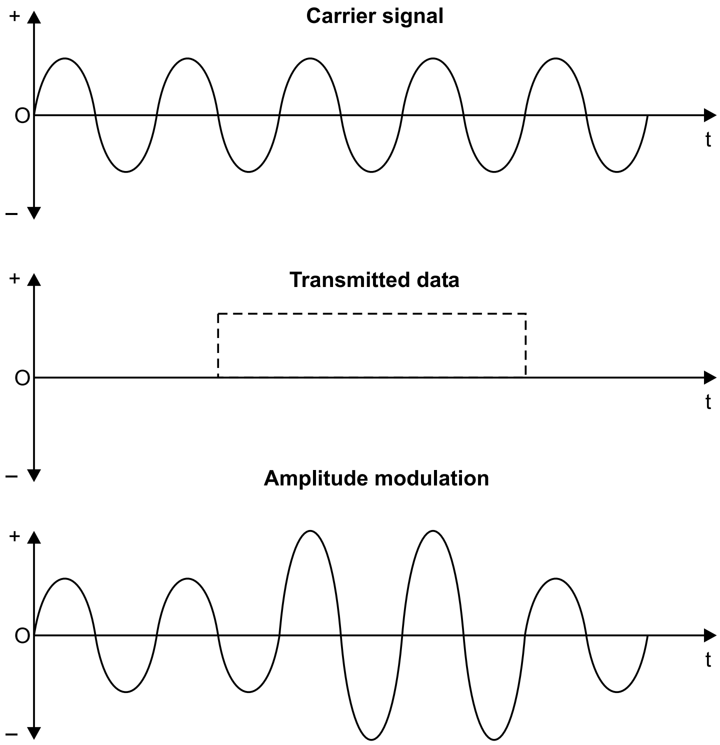


Fig. 2.1 Basic amplitude modulation waveforms

In practice, the carrier may be high frequency (HF), while the modulation is audio 1.

Let the carrier signal and the modulating signal a_c and a_m respectively be as,

$$a_c = A_c \sin(\omega_c t + \theta_c)$$

$$a_m = A_m \sin(\omega_m t + \theta_m)$$

Note that phase angle has been ignored in both expressions, since it is unchanged by the amplitude modulation process. [It is not possible to ignore phase angle in frequency and phase modulation.]

From this, you can see that the maximum amplitude AC of the unmodulated carrier will have to be made proportional to the instantaneous modulating amplitude $A_m \sin \omega_m t$ when the carrier is amplitude modulated.

AM is generally familiar as an entertainment radio broadcast transmission technique. The receiver is tuned to the constant carrier frequency of a particular station. The transmitter modulates or produces variations in the amplitude of a signal which can be detected by the receiver to reproduce the original information.

2.3 AM BANDWIDTH REQUIREMENT

Bandwidth

Bandwidth refers to the range of transmission frequencies that are carried on a communication line.

Bandwidth is actually computed by substracting the lowest frequency found in the signal from the highest frequency found in that signal, such as a telephone channel or a voice transmission.

It is reasonable to expect that the frequency range (Bandwidth) required for a given transmission should depend on the bandwidth occupied by the modulation signals themselves.

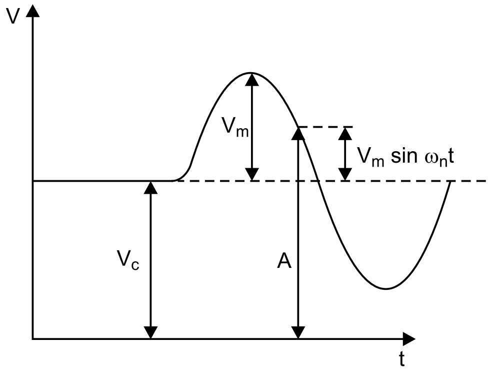


Fig. 2.2 Distorted waveform

A high-fidelity audio signals requires a range of 50 to 15000 Hz, but a bandwidth of 300 to 3400 Hz is adequate for a telephone conversation. When a carrier has been similarly modulated with each, a greater bandwidth will be required for the high-fidelity (Hi-Fi) transmission.

i.e. the transmitted bandwidth need not be exactly the same as the bandwidth of the original signals.

Let V_c and V_m be the instantaneous values of the carrier signal voltage and modulating signal voltage respectively, and is given by

$$V_c = V_c \sin \omega_c t \qquad \text{...(I)}$$

$$V_m = V_m \sin \omega_m t \qquad \text{...(II)}$$

Note that the phase angle has been ignored in both expressions, since it is unchanged by the modulation process.

From the definition of AM, it follows that the amplitude V_c of the carrier signal will have to be made proportional to instantaneous modulating voltage $V_m \sin \omega_m t$. When the carrier is amplitude modulated.

Here we know that the frequencies present in the amplitude modulating wave are the carrier frequency and the first pair of side band frequencies.

Where sideband frequencies are as

$$f_{SB} = f_c \pm n f_m \quad \because \text{ in the 1}^{st} \text{ pair } n = 1 \qquad \text{...(III)}$$

$$\therefore \qquad f_{SB} = f_c \pm f_m$$

where a carrier is amplitude modulated, the proportionality constant made equals to unity.

When their is temporarily modulation occurs, the amplitude of the carrier is varied by its instantaneous value, and when there is no modulation the amplitude of the carrier is equal to its unmodulated value.

Figure 2.2 shows some distortion will occur if V_m is greater than V_c. Distortion is due to overriding of amplifier stage.

$\therefore$ The ratio $\frac{V_m}{V_c}$ occurs and that is also called as **modulation index,** which is as

$$m = \frac{V_m}{V_c} \qquad \text{...(IV)}$$

The modulation index is a number lying between 0 – 1, and it is also called the **percentage modulation.**

From equations (II) and (III), the amplitude of amplitude modulated voltage is,

$$\begin{aligned} A &= V_c + V_m \\ &= V_c + V_m \sin \omega_m t \\ &= V_c + m V_c \sin \omega_m t \end{aligned}$$

$$= V_c(1 + m \sin \omega_m t) \qquad \text{...(V)}$$

Also,

The instantaneous voltage of the resulting AM is

$$V = A \sin \theta$$

$$V = A \sin \omega_c t$$

$$\therefore \quad V = V_c(1 + m \sin \omega_m t) \sin \omega_c t$$

$$\therefore \quad V = V_c \sin \omega_c t + \frac{mV_c}{2}\cos(\omega_c - \omega_m)t - \frac{mV_c}{2}\cos(\omega_c + \omega_m)t$$

...(VI)

*[Expanded by using trignometrical relation sin x sin y = cos (x – y) – cos (x + y)/2]

From equation (VI), we conclude that, it has three terms.

1. $V_c = V_c \sin \omega_c t = V_c \sin 2\pi f_c t$

→ This represents original carrier signal.

2. $\frac{mV_c}{2}\cos(\omega_c - \omega_m)t = \frac{mV_c}{2}\cos 2\pi(f_c - f_m)t$

→ This represents the lower sideband frequency i.e. (LSB).

3. $\frac{mV_c}{2}\cos(\omega_c + \omega_m)t = \frac{mV_c}{2}\cos 2\pi(f_c + f_m)t$

→ This represents the upper sideband frequency i.e. (USB).

∴ We can write

$$\text{L.S.B.} = f_c - f_m \qquad \text{...(VII)}$$

$$\text{U.S.B.} = f_c + f_m \qquad \text{...(VIII)}$$

The frequency of AM wave contains three discrete frequencies as shown in Fig. 2.3.

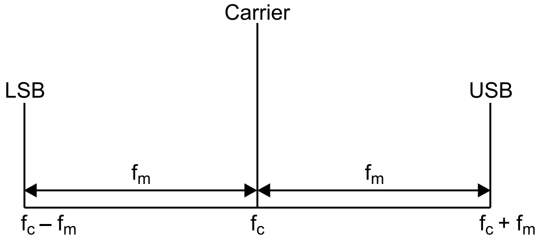


Fig. 2.3 Frequency spectrum

In these three frequencies

Central frequency = f_c ;

$$\text{L.S.B. frequency} = f_c - f_m \;;$$

$$\text{U.S.B. frequency} = f_c + f_m.$$

$$\therefore \quad \text{Bandwidth} = \text{U.S.B.} - \text{L.S.B.}$$

$$\therefore \quad \text{B.W.} = (f_c + f_m) - (f_c - f_m)$$

$$\therefore \quad \text{B.W.} = 2f_m \qquad \text{...(IX)}$$

$\therefore$ From equation (IX), we conclude that, the "Bandwidth required for amplitude modulation is twice the frequency of the modulating signal".

OR

In amplitude modulation broadcasting service where several sine waves simultaneously occurs, "the Bandwidth required is twice the highest modulating frequency".

2.4 FREQUENCY MODULATION

Frequency modulation may be defined as a technique for slightly modifying the basic carrier frequency of a signal, in a way that superimposes additional information that can be recovered by a receiver.

Here the amplitude remains constant and the frequency changes. In this modulation we could use several frequencies rather than just the f_1, f_2 and so on.

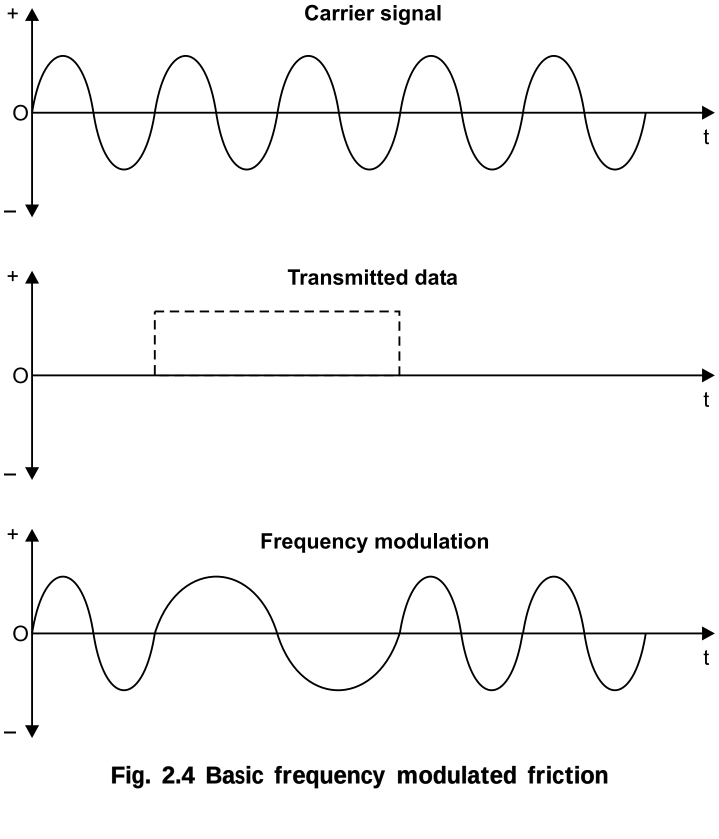


Fig. 2.4 Basic frequency modulated friction

A binary 1 is represented by one frequency and a binary 0 is represented by another frequency.

An example of entertainment radio for analog transmission of analog data is Entertainment Radio.

In this case the basic carrier frequencies tend to be much higher so that the modulating variations are very small by comparison. The FM band is in the 100 MHz. (One million cycles/sec.).

For digital data, only two variations on the basic carrier frequency are needed either the AM band or FM band of carrier frequency would be useful for FM transmission.

FM can be more expensive/(costly), if the higher carrier frequencies are used and because of the sensitivity required for the receiver to detect the variations properly.

2.5 FM BANDWIDTH REQUIREMENT

Frequency modulation is a system in which the amplitude of the modulated carrier is kept constant, while its frequency is varied by the modulating signal.

The general equation of an unmodulated wave, or carrier can be written as

$$x = \mathrm{A}\sin(\omega t + \theta) \qquad \text{...(1)}$$

where

x = instantaneous values of current and voltage.

A = maximum (amplitude) of current and voltage.

ω = angular frequency, rad/s.

θ = phase angle, rad.

If the frequency of the carrier is made to vary, frequency modulated waves are obtained.

By the definition of frequency modulation, the amount by which the carrier frequency is varied from its unmodulated value is called the **deviation,** is made proportional to the instantaneous value of the modulating voltage.

The rate at which this frequency variation takes place is naturally equal to the modulating frequency.

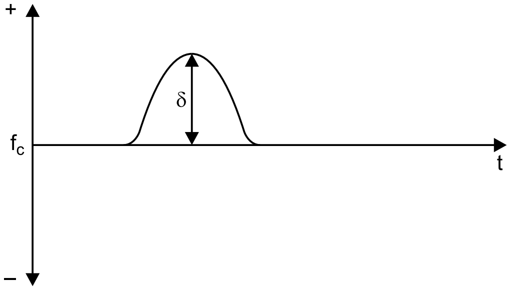


Fig. 2.5 Frequency Vs time in FM

Figure 2.5 shows that the instantaneous frequency f_1 of the modulated frequency wave is,

$$f = f_c(1 + k\text{V}_m \cos \omega_m t) \qquad \text{...(2)}$$

where

$$f_c = \text{unmodulated carrier frequency ; Hz.}$$

$$\text{V}_m \cos \omega_m t = \text{instantaneous value of modulating voltage ; V.}$$

The maximum deviation from this particular signal occurs when the cosine term has its maximum value. i.e. ± 1. Under these conditions, the instantaneous frequency is,

$$f = f_c(1 \pm k\text{V}_m) \qquad \text{...(3)}$$

so that the maximum deviation is

$$\delta = k\text{V}_m f_c \qquad \text{...(4)}$$

The instantaneous amplitude of FM signal is of the form,

$$ef_m = \text{A sin [F}(\omega_c, \omega_m)] = \text{A sin } \theta \qquad \text{...(5)}$$

where $\text{F}(\omega_c,\omega_m)$ is a function of the carrier and modulating frequencies.

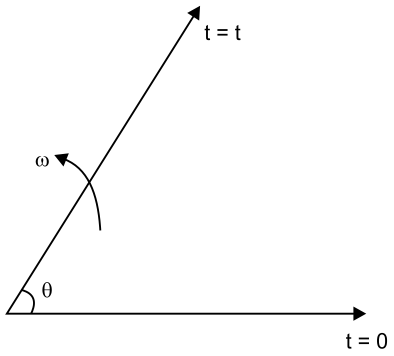


Fig. 2.6 Frequency modulated signals

Figure 2.6 shows, θ is the angle traced out by phasor A in time $t = t$. If A were rotating at a constant velocity, say P angle θ = Pt. However, the velocity P is not constant, but is variable and is governed by angular frequency $\omega = 2\pi f$ and using equation (2).

$$\omega = \omega_c(1 + k\text{V}_m \cos \omega_m t) \qquad \text{...(6)}$$

therefore,

$$\theta = \int \omega \delta t$$

$$= \int \omega_c (1 + k\text{V}_m \cos \omega_m t)dt$$

$$= \omega_c \left[t + \frac{k\text{V}_m \sin \omega_m t}{\omega_m} \right]$$

$$= \omega_c t + \frac{k\mathrm{V}_m \sin \omega_m t}{\omega_m}$$

$$= \omega_c t + \frac{k\mathrm{V}_m f_c \sin \omega_m t}{f_m}$$

$$= \omega_c t + \frac{\delta}{f_m} \sin \omega_m t \qquad ...(7)$$

Using equation (7) equation (5) can be written as,

$$ef_m = \mathrm{A} \sin \left[\omega_c t + \frac{\delta}{f_m} \sin \omega_m t \right] \qquad ...(8)$$

The modulation index for FM is defined as

$$mf = \frac{\text{max. frequency deviation}}{\text{modulating frequency}} = \frac{\delta}{f_m} \qquad ...(9)$$

$\therefore$ substituting equation (9) in equation (8), we obtain,

$$ef_m = \mathrm{A} \sin (\omega_c t + mf \sin \omega_m t) \qquad ...(10)$$

It should be noted that, as the modulating frequency, f_m decreases and δ remains constant the m_f increases.

FM on the other hand contains many successive sideband is displaced from the next adjacent band by frequency of modulating signal. The No. of sidebands and the **signal strength** of each sideband is determined by the **modulation** index.

Mathematically the sidebands are defined by set of tables or curves known as **Bessel functions** shown in figure 2.7. These functions are plotted relative to the amplitude of the carrier and the trace of relative amplitude of each sideband and carrier for different value of modulation index.

Each of the sidebands whose amplitudes are determined by the curves shown in figure present a wave shape in the form of a damped oscillation, which is a function of their instantaneous modulation index. The carrier signal represented by J_o actually disappears at specific high values of m_f. At a particular value of m_f, the energy will be distributed over the many sidebands in a manner indicated in the chart.

FM also differs from AM with regards to variation in total transmitted power with modulation. AM increases total-power, transmitted power with a modulation up to a maximum increase of 50%. During the process, the carrier power remains constant and the 50% increase appears as sideband power. In the FM wave, the total power remains constant, but the distribution of energy over the sidebands varies with modulation.

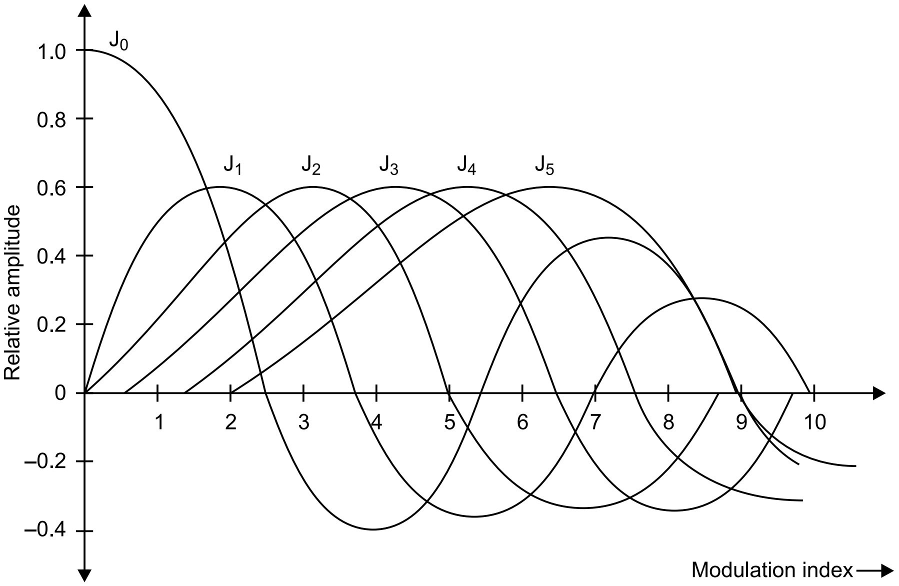
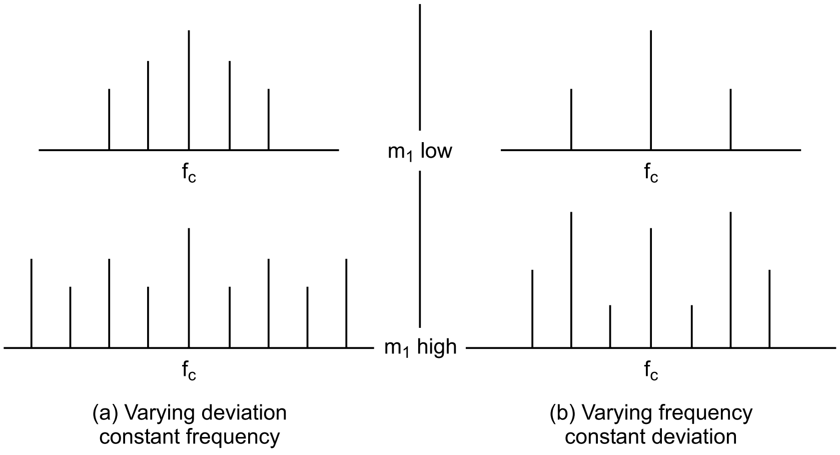


Fig. 2.7 Bessel functions

The modulation index (m_f) for FM waves varies directly with the deviation δ and inversely with amplitude of modulating signal f_m. Therefore, m_f can be varied by holding the frequency constant and bands that are frequency spaced by the same amount, but the amplitudes (any extent) of the sidebands differ. This is shown in Fig. 2.8(a).

Another, possibility for m_f variation is to keep the deviation constant and vary the frequency of the modulating signal.



(a) Varying deviation constant frequency

(b) Varying frequency constant deviation

Fig. 2.8 FM frequency

This results in sidebands that are spaced differently as shown in figure 2.8(b). In practice, both the amplitude and frequency of the modulating signal are varying and the result is a complex FM. Waveshape described mathematically by Bessel functions.

The bandwidth required for an FM system can be calculated after referring to the Bessel function tables and deciding which is the highest J term to be included for a particular modulation index. A rough approximation of required sideband is the sum of deviation plus highest modulating frequency. The bandwidth then becomes this number doubled.

A more specific definition of this relationship is listed in the Table 2.1.

FM broadcast standards specify a maximum deviation of 75 kHz and a maximum modulating frequency of 15 kHz. This equates to an m_f of 75/15 or 5.

Table 2.1 Bandwidth required for a FM signal

Modulation index (M_f)	*Number of significant sidebands*	*Bandwidth*	
		As multiple of modulating frequency	*As multiple of frequency deviation*
0.1	2	2	20
0.5	4	4	8
1	6	6	6
2	8	8	4
5	16	16	3.2
10	28	28	2.8

The frequency components of an FM signal extend to infinity on either side of the carrier frequency. However their amplitudes fall off rapidly outside a certain frequency range. According to **Carson's rule**, the approximate value of Bandwidth of an FM signal is :

Bandwidth, **BW** $= 2(\delta + f_h)\text{Hz}$ ...

where

δ = maximum deviation in the carrier from its unmodulated frequency.

f_h = highest frequency in the message signal.

Frequency Modulation requires a greater bandwidth than amplitude modulation. It is used for both land line telemetering and radio frequency telemetering.

Problem 1. ***A 1000 kHz carrier is simultaneously modulated with 300 Hz, 800 Hz and 2 kHz audio sine waves. What will be the frequencies present in the output?***

Solution.

Carrier frequency $f_c = 1000\ k\text{Hz}.$

Three modulating signals having frequencies.

$f_{m1} = 300\text{ Hz}.$

$f_{m2} = 800\text{ Hz}.$

$$f_{m3} = 2 \text{ kHz.}$$

Here three sideband frequencies presents on accounts of the modulating signals. Therefore, they have three LSB frequencies and three USB frequencies and these are:

$$\begin{aligned} & & & \text{L.S.B.} \quad \text{U.S.B.} \\ f_c \pm f_{m1} &= 1000 \pm 0.3 &=& \mathbf{999.7}\boldsymbol{k} \ ; \ \mathbf{1000.3}\boldsymbol{k} \\ f_c \pm f_{m2} &= 1000 \pm 0.8 &=& \mathbf{999.2}\boldsymbol{k} \ ; \ \mathbf{1000.8}\boldsymbol{k} \\ f_c \pm f_{m3} &= 1000 \pm 2 &=& \mathbf{998}\boldsymbol{k} \ ; \ \mathbf{1002}\boldsymbol{k} \end{aligned}$$

Problem 2. ***The tuned circuit in a simple AM transmitter uses a 50 μH inductance and 1nF capacitance. If the oscillator output is modulated by radio frequencies up to 10 kHz. What is the Bandwidth occupied by sidebands?***

Solution. Here Given data :

$$\begin{aligned} \text{L} &= 50 \ \mu\text{H} \\ \text{C} &= 1\text{nF} \\ f_m &= 10 \text{ kHz.} \end{aligned}$$

$$\text{Carrier frequency } f_c = \frac{1}{2\pi\sqrt{\text{LC}}}$$

$$\therefore \quad f_c = \frac{1}{2\pi\sqrt{50\times10^{-6}\times1\times10^{-9}}}$$

$$\therefore \quad f_c = 712 \times 10^3 \text{ Hz}$$

$$\therefore \quad \boldsymbol{f_c} = \mathbf{712 \ kHz}$$

Since the highest f_m is 10 kHz

$$\begin{aligned} \therefore \quad \text{L.S.B.} &= f_c - f_m \\ \therefore \quad \text{L.S.B.} &= 712 - 10 \\ \therefore \quad \text{L.S.B.} &= 702 \text{ KHz} \end{aligned}$$

AND

$$\begin{aligned} \text{U.S.B.} &= f_c + f_m \\ &= 712 + 10 \\ \mathbf{U.S.B.} &= \mathbf{722 \ kHz} \end{aligned}$$

$$\begin{aligned} \therefore \quad \text{Bandwidth} &= \text{U.S.B.} - \text{L.S.B.} \\ \text{B.W.} &= 722 - 702 \end{aligned}$$

$$\boxed{\mathbf{B.W. = 20 \ kHz.}}$$

Problem 3. ***In an FM system, when the audio frequency (A.F.) is 500 Hz and the A.F. voltage is 2.4 V, the deviation is 4.8 kHz. If the A.F. Voltage is now increased to 7.2 V. What is the deviation? If the A.F. voltage is raised to 10 V while its frequency is decreased, to 200Hz, what is the deviation? Find the modulation index in each case.***

Solution. Here given data

$$\delta_1 = 4.8 \text{ kHz}$$

$$V_{m_1} = 2.4V$$

$$AF = f_{s1} = 500 \text{ Hz} \; ; \; f_{s3} = 200 \text{ Hz.}$$

The maximum deviation is given by $\delta = kV_m f_c$.

Here f_c is constant

$$\therefore \quad \delta_1 \propto V_m$$

$$\delta_1 = k^1 V_{m1}$$

For $\delta_1 = 4.8$ kHz, $V_{m1} = 2.4V$

$$\therefore \quad k^1 = \frac{4.8}{2.4}$$

$$\therefore \quad k^1 = \mathbf{2 \text{ kHz/V.}}$$

For $V_{m2} = 7.2V$

$$\delta_2 = K^1 V_m$$

$$\therefore \quad \delta_2 = 2 \times 7.2$$

$$\therefore \quad \delta_2 = \mathbf{14.4 \text{ kHz.}}$$

For $V_{m3} = 10V$

$$\therefore \quad \delta_3 = 2 \times 10$$

$$\therefore \quad \boxed{\boldsymbol{\delta_3 = 20 \text{ kHz.}}}$$

The modulation indices for different signal frequencies are.

$$\therefore \quad mf_1 = \frac{\delta_1}{f_{S1}}$$

$$\therefore \quad mf_1 = \frac{4.8}{500 \times 10^{-3}}$$

$$\therefore \quad \boxed{\boldsymbol{mf_1 = 9.6}}$$

$$mf_2 = \frac{\delta_2}{f_{S1}}$$

$$\therefore \quad mf_2 = \frac{14.4}{500 \times 10^{-3}}$$

$$\therefore \quad \boxed{\boldsymbol{mf_2 = 28.8}}$$

$$mf_3 = \frac{\delta_3}{f_{S3}} = \frac{20}{200 \times 10^{-3}}$$

$$\therefore \quad \boxed{\boldsymbol{mf_3 = 100}}$$

Problem 4. ***Find the carrier and modulating frequency. The modulation index and the maximum deviation of the FM wave represented by***

$$ef_m = 12 \sin (6 \times 10^8 t + 5 \sin 1250\ t)V$$

Solution. The FM is given as

$$ef_m = 12 \sin (6 \times 10^8\ t + 5 \sin 1250\ t)\ \text{V} \quad \text{...(I)}$$

The equation for instantaneous voltage of FM is as

$$ef_m = A \sin (\omega_c t + mf \sin \omega_m t) \quad \text{...(II)}$$

Compairing these two equations;

$$\omega_c = 6 \times 10^8 \text{ rad/s}$$

$$\therefore \quad f_c = \frac{\omega_c}{2\pi} = \frac{6 \times 10^8}{2\pi}$$

$$\therefore \quad \boxed{f_c = 95.5 \text{ MHz.}}$$

Also $\omega_m = 1250$ rad/s

$$\therefore \quad f_m = \frac{1250}{2\pi}$$

$$\therefore \quad \boxed{f_m = 199 \text{ Hz.}}$$

$\therefore$ Modulation index $\boxed{m_f = 5}$

$\therefore$ Maximum deviation $\delta = m_f f_m$

$$\therefore \quad \delta = 5 \times 199$$

$$\therefore \quad \boxed{\delta = 995 \text{ Hz.}}$$

K.M.S value of voltage is, (For FM)

$$\therefore \quad P = \frac{\left(12\sqrt{2}\right)^2}{10}$$

$$\therefore \quad \boxed{P = 7.2 \text{ W.}}$$

Problem 5. ***What is the bandwidth required for an FM signal in which the modulating frequency is 2 kHz and the maximum deviation is 10 kHz.***

Solution. The modulation index

$$mf = \frac{\delta}{f_m}$$

$$\therefore \quad mf = 5$$

The bandwidth as multiple of frequency of modulating frequency signal is 16.

$$\therefore \quad \text{B.W.} = 16\ f_m \qquad 2(\delta + f_h)$$

$$\therefore \quad \text{B.W.} = 16 \times 2 \qquad 2(10 + 2) = 24 \text{ kHz.}$$

$$\therefore \quad \boxed{\text{B.W.} = 32 \text{ kHz.}}$$

2.6 CONCEPT OF NOISE

NOISE

Noise is unwanted signal from sources other than the transmitted signal. i.e. it is a signal that does not convey any information.

Particularly noise can be divided into two subgroups.

1. External Noise.
2. Internal Noise.

1. External Noise

The noise which is created or caused outside the communication or receiver system is called an external noise. External noises are somewhat uncontrollable and these are,

(*a*) Atmospheric noise.

(*b*) Extra-terrestrial/space noise.

(*c*) Man-made or individual noise.

(a) Atmospheric Noise

It is caused by lighting discharge in thunderstorm and other natural disturbance in atmosphere. These noise impulse are electric in nature and spread over the complete frequency spectrum which is used for radio communication. These noise impulse get propagated over the earth in the same fashion as the radio waves of same frequencies. The receiving antenna not only pick up the desired signal but also the noise from thunderstorm and various disturbance causes at the output. The field strength of atmospheric noise varies inversely proportional to frequency. Thus large atmospheric noise is generated in low or medium frequency band while very little noise is generated in very high frequency band.

(b) Extra-Terrestrial/Space Noise

Space noise is divided into two categories i.e.

1. Solar noise.
2. Cosmic noise.

1. **Solar noise:** This is electrical noise generated from the sun. There is continuous radiation from sun i.e. result from such which is large body of very high temperature (6000°C) radiate electrical energy spectrum which is in the form of noise which spread over all the spectrum used for radio communication.
2. **Cosmic noise:** Distant, stars also sun and have high temperature. These stars therefore, radiate the noise in the same way as sun. The noise receive from the distant, star is known as **thermal noise** and distributed almost uniformly over the entire and almost effects on communication of radio waves.

(c) Man-made/Industrial Noise

It is the electric noise produced by such source like automobiles. A aircraft ignition, electric motors and switch gear leakage from higher voltage light.

Fluorescent light and many of man-made noise like electrical machine are most intensive in industrial area and populated urban area.

2. Internal Noise

The noise which is generated inside the communication system is an internal noise. These are produced by properly design of receiver circuitary. These are:

(*a*) Thermal Noise.

(*b*) Shot Noise.

(*c*) Transit-time Noise.

(a) Thermal Noise

Conductor contains large No. of electrons and ions which are bound by molecular forces. These are vibrating about their normal position. These vibrations depends on temperature. The movement of electrons and ions constitute current which over a long time average to 0. But this random gives a voltage called thermal/white or Johnson's noise voltage.

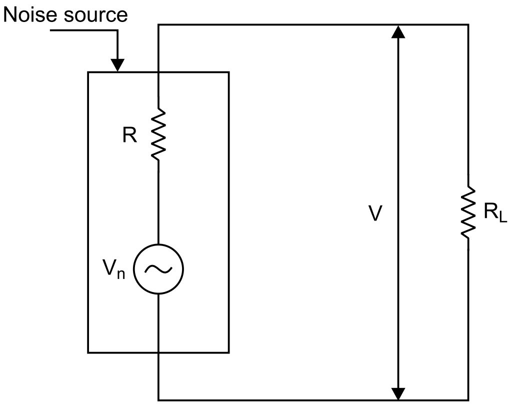


Fig. 2.9 Thermal noise source

Thermal noise is generated in resistors or in resistive components due to rise in temperature. The noise generated by resistor is directly proportion to absolute temperature and bandwidth.

$\therefore$ $\quad P n \propto T$

and $\quad P n \propto B$

$P n \propto TB$

$\therefore$ $\quad P n = KTB \qquad \text{... (1)}$

where $\quad P n$ = Maximum noise power generated.

K = Boltman's const. (1.38 × 10^{23}J/kelvin).

T = Absolute temperature.

B = Bandwidth over which noise is generated.

$$V = \frac{V_n}{R} \text{ and } Pn = \frac{V^2}{R}$$

$$\therefore \quad Pn = \frac{V_n^2}{4R}$$

$$\therefore \quad V_n^2 = 4RPn \qquad \text{... (2)}$$

Putting P*n* = KTB in equation (2), we get

$$V_n^2 = 4RKTB$$

$$\therefore \quad V_n = \sqrt{4RKTB}$$

$$\therefore \quad \mathbf{V_n = } 2\sqrt{RKTB}$$

This RMS noise voltage can only be measured by electric voltmeter for few volt amplitude. Thermal noise can be ignored. But for weak signal, there is thermal noise which an important measurement. Their is danger of amplifying the noise along with the signal.

(b) Shot Noise

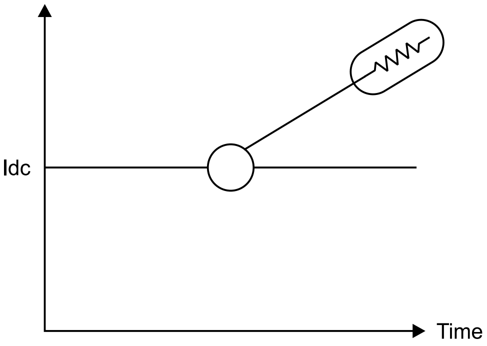


Fig. 2.10 Shot noise source

It is due to amplifying active device only also it is caused by discrete nature of matter. If average flow of charge carriers in any device is constant then average number of charge carriers/unit time are found constant. But there is always fluctuations about this average. This fluctuation depends on random emission of charge carriers from cathode. This fluctuations falls random having noise current superimposes on direct current of a device. After amplification this noise on slide, leads shot following on metal surface. Hence this is a shot noise.

There are number of variables generating this noise. So there is calculation by approximation statistical equation.

$$i_n = \sqrt{2ei\text{PB}}$$

where i_n = Noise current

i_p = Direct diode current

e = Charge on element

B = Bandwidth

It is difficult to ratio of current to thermal noise voltage for this equivalent noise resistor is considered and device is considered noiseless.

(c) Transit-time Noise

Transit-time noise occurs in very high frequency range i.e. the time by an electron to travel from emitter to collector in transistor which becomes comparable to periodic time of signal being amplified. As a result of this input admitance of transistor increases and thermal noise is generated. In this input which represents noise contribution of transit time.

2.7 NOISE FIGURE AND NOISE TEMPERATURE

NOISE FIGURE

Noise Factor

Noise factor 'F' is defined as ratio of signal to noise power at input to signal to noise power at output.

$$\text{F} = \frac{\text{S/N at input}}{\text{S/N at output}}$$

S–Signal

N–Noise

$$\therefore \quad nf = 10 \log \text{F}$$

$$\text{Noise} = 10\log\left[\frac{\text{O/P noise voltage with noise at I/P}}{\text{O/P noise voltage without noise at I/P}}\right]$$

Signal to noise ratio

It is ratio of signal power to noise power. The noise factor measurements are important because they are a major of noise added to a signal by a device in the measurement system. If noise factor is expressed in decimal it is known as noise figure.

$$\textbf{Noise Figure} = \textbf{10 log F} \quad \text{...(1)}$$

The measurement of noise figure is most meaningful measurements for amplifiers, transistors and vaccum tubes since it is a major of the noise generated within the device, the measurement of noise figure requires use of known source of noise. The noise figure from equation (1) is,

$$nf = 10\log\left[\frac{\text{noise voltage at O/P with noise at I/P}}{\text{noise voltage at O/P without noise at I/P}}\right]$$

$$= 10\log\left[\frac{V_n}{V_o}\right]$$

where,

Vn = Is the output noise voltage with noise source injected into input.

V_o = Is the output noise voltage without noise at input.

Difference between the two voltages represents the noise added by the device under test. For **ideal case** i.e. no noise condition, noise figure equals to 1.

Noise figure may be expressed in actual ratio form of dB scale and is given by the expression.

$$\boxed{F = 1+\frac{\text{Req}^1}{\text{Rq.}}}$$

NOISE TEMPERATURE

Noise figure is a good indicator of a noise measurement but while considering physical devices or physical antenna and microwave low noise antenna or receive the noise figure is not convenient to measure noise at that time noise temperature is extensively used for antenna and low microwave amplifiers. The concept of noise temperature is similar to noise power so that the total noise power from many sources is,

$$P_t = \bar{k}T_t B \quad ...(1)$$

$$P_t = P_1 + P_2$$

$$\therefore \quad P_t = \bar{k}BT_1 + \bar{k}BT_2$$

$$\therefore \quad P_t = \bar{k}B(T_1+T_2) \quad ...(2)$$

compairing equations (1) and (2), we get

$$\boxed{T_t = (T_1 + T_2)}$$

where

P_1 and P_2 = are the noise received by the antenna and noise generated by antenna.

T_1 and T_2 = are the individual noise temperature corresponding to P_1 and P_2.

T_t is the total noise temperature.

As the term noise equivalent noise temperature is concerned. The **equivalent noise temperature teq.**

For receiver assumed that

$$Req^1 = R_a$$

If assumption is correct the Req must be at temperature other than that of all the components including R_a.

The noise figure F and T equivalent is written as,

$$F = 1+\frac{Teq}{To}$$

$$\therefore \qquad \boxed{Teq = To[F - 1]}$$

where

Teq is the equivalent noise temperature of receiver.

$$To = 27°C + 273 = 300°K$$

F is noise.

By knowing noise figure, it is easy to calculate equivalent noise temperature, teq.

QUESTIONNAIRES

1. Define modulation. List the types of modulation.

   Represent graphically, (in AM) when

   Modulation index < 1

   Modulation index = 1

   Modulation index > 1
2. Why we need modulation? Explain frequency modulation.
3. Explain frequency modulation. Also compare with amplitude modulation.
4. Explain FM detection process with waveforms.
5. Explain in detail various types of noise.
6. Explain noise figure.
7. Define following:
   - Bit rate
   - Baud rate
   - Noise figure
   - Noise temperature
   - Shot noise
8. Calculate the noise voltage at the input of a RF amplifier using a device that has a 300Ω equivalent noise resistance and a 300Ω input resistance. The bandwidth of the amplifier is 5 MHz and temperature is 17°C.

Chapter 3

MULTIPLEXING

INTRODUCTION

All the transmission media's have capacities great enough to carry more than one voice channel. In other words, their bandwidths are considerably greater than the 3kHz needed for transmitting the human voice. At the top end of the scale, microwave and fiber-optic circuits carry thousands of voice channels, at the lower end of the scale, each voice channel may be split into 12 or 24 telegraph channels.

Where a facility is set up, such as a chain of microwave links, which has a broad bandwidth it is very desirable to make the maximum use of this bandwidth by making it carry as many channels as possible. It is often desirable to construct a communication link with as wide a bandwidth as possible and then divide the bandwidth between as many users as possible. Many separate signals are multiplexed together so that they can travel as one signal over a high bandwidth.

In a multiplex system, two or more signals are combined so that they can be transmitted together over one physical cable or radio link. So the word multiplexing, in general, "means the use of one facility to handle several separate but similar operations simultaneously".

3.1 CONCEPT OF MULTIPLEXING

MULTIPLEXING

It is a process in which several independent signals combine into a form that can be transmitted across a communication links and then separated into its original components.

In data transmission, a function that permits two or more data sources to share a common transmission medium such that each data source has its own channel.

A common application of multiplexing is in long-haul communications. Trunks on long-haul networks are high-capacity fiber, co-axial or microwave links. These links carry large number of voice and data transmissions simultaneously using multiplexing.

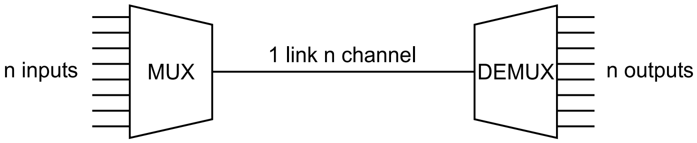


Fig. 3.1 Multiplexer

Figure 3.1 shows multiplexing function in its simplest form. There are *n* inputs to a multiplexer. The multiplexer is connected by a single data link to a demultiplexer. The link is able to carry *n* separate channels of data. The multiplexer combine (Multiplexes) data from the *n* input lines and transmits over a higher capacity data link. The demultiplexer accepts the multiplexed data stream, separate (demultiplexes) the data according to channel and delivers them to the appropriate output lines.

Uses of Multiplexing (In data communication)

1. The higher the data rate, the more cost-effective the transmission facility, *i.e.*, for a given application and over a given distant the cost/kbps declines with an increase in the data rate of the transmission facility. Similarly, the cost of transmission and receiving equipments, per kbps, declines with increasing data rate.
2. Most individual data-communicating devices requires relatively modest data-rate support.

   For most terminal and personal computer applications, a data rate of between 9600 bps and 64 kbps is generally adequate.

   [These statements were phrased in data communication, also apply to voice communication].

3.2 FREQUENCY DIVISION MULTIPLEXING

A technique for sharing a common transmission medium by dividing the frequency range carried by the medium into multiple channels, so that modulation of a carrier frequency within one channel does not interfere with signals in any other channel.

FDM is possible when the useful bandwidth of the transmission medium exceeds the required bandwidth of signals to be transmitted. A Number of signals can be carried simultaneously if each signal is modulated onto a different carrier frequency and the carrier frequencies are sufficiently separated that the bandwidths of the signals do not overlap.

A general case of FDM is shown in the figure 3.2. Six signal sources are fed into a multiplexer which modulates each signal onto a different frequency (f_1-----f_6). Each modulated signal requires a certain bandwidth centred around its carrier frequency referred to as a channel. To prevent interference, the channels are separated by guard bands, which are unused portions of the spectrum.

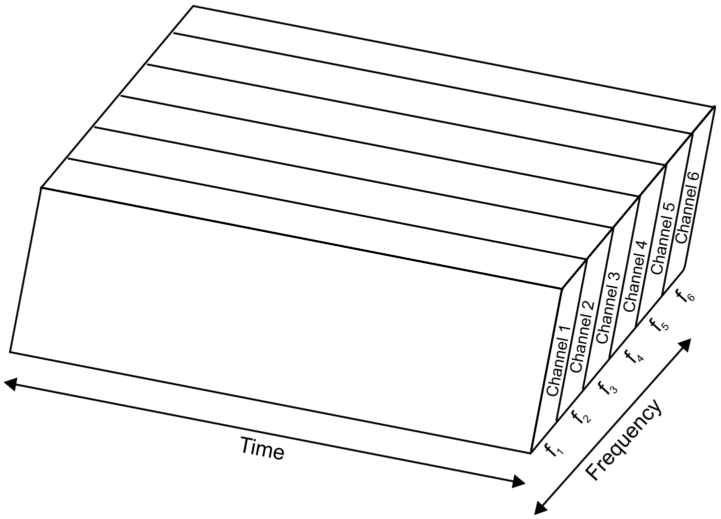


Fig. 3.2 FDM

A familiar example of FDM is broadcast and cable television.

A generic depiction of an FDM system is shown in figure 3.3. A number of analog and digital signals [m_i (t), i = 1, N] are to be multiplexed onto the same transmission medium. Each signal $m_i(t)$ is modulated onto a carrier f_{sci}; because multiple carriers are to be used, each is referred to as a subcarrier. Any type of modulation may be used. The resulting modulated analog signals are then summed to produce a composite signal $m_c(t)$. Figure shows the result. The spectrum of signal $m_i(t)$ is shifted to be centered on f_{sci}. For this scheme to work, f_{sci} must be chosen so that the bandwidths of the various signals do not overlap ; otherwise, it will be impossible to recover the original signals.

The composite signal may then be shifted as a whole to another carrier frequency by an additional modulation step. We will see examples of this below. This second modulation step need not use the same modulation technique as the first.

The composite signal has a total bandwidth B, where $\mathrm{B} > \sum_{i=1}^{\mathrm{N}} \mathrm{B}_{sci}$. This analog signal may be transmitted over a suitable medium. At the receiving end, the composite signal is passed through N bandpass filters, each filter centered on f_{sci} and having a bandwidth B_{sci}, for 1 < i < N, in this way, the signal is again split into its component parts. Each component is then demodulated to recover the original signal.

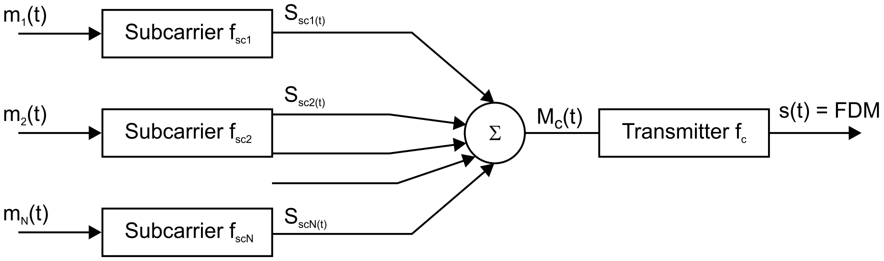


Fig. 3.3 (a) Transmitter

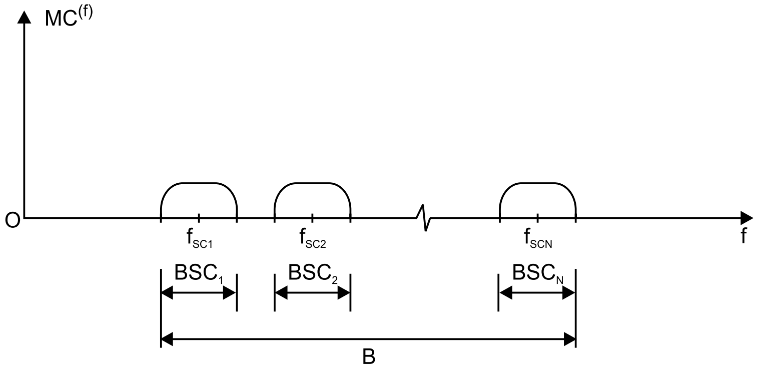


Fig. 3.3 (b) Spectrum of composite signal (+f)

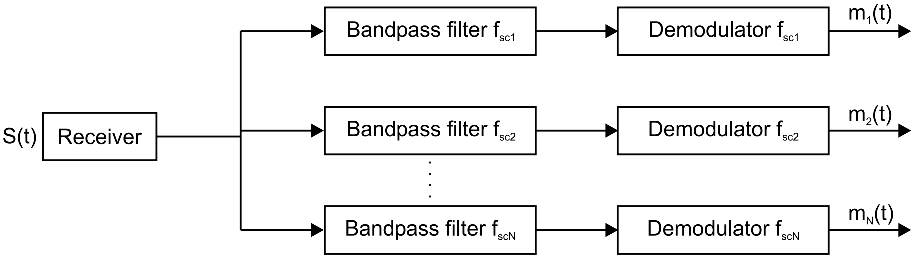
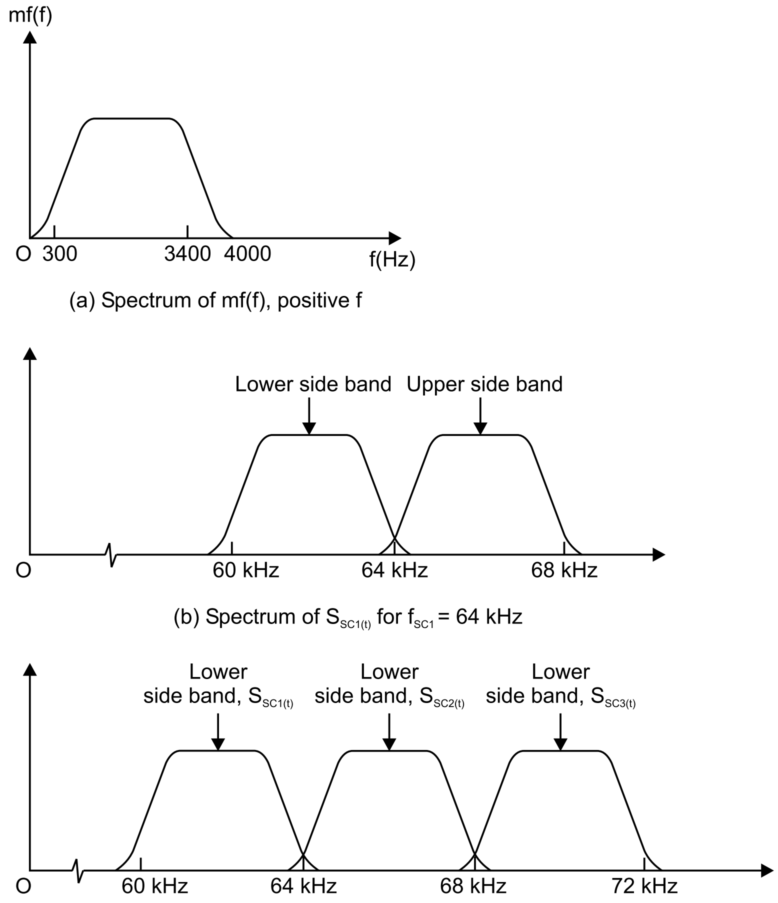


Fig. 3.3 (c) Receiver

Simple example

Let us consider a simple example of transmitting three voice signals simultaneously over a medium. As was mentioned, the bandwidth of a voice signal is generally taken to be 4 kHz, with an effective spectrum of 300 to 3400 Hz. Which is shown in figure 3.4 (*a*). If such a signal is used to amplitude-modulate a 64 kHz carrier, the spectrum of figure 3.4 (*b*) results. The modulated signal has a bandwidth of 8 kHz, extending from 60 to 68 kHz. To make efficient use of bandwidth, we elect to transmit only the lower sideband.

Now, if three voice signals are used to modulate carriers at 64, 68 and 72 kHz, and only the lower sideband of each is taken, the spectrum of figure 3.4(*c*) results.



(a) Spectrum of mf(f), positive f

(b) Spectrum of $S_{SC1(t)}$ for f_{SC1} = 64 kHz

(c) Spectrum of composite signal using subcarriers at 64 khz, 68 khz and 72 khz

Fig. 3.4 FDM of three voice band signals

This figure points out two problems that an FDM system must cope with. The first is **crosstalk**, which may occur if the spectra of adjacent component signals overlap significantly. In the case of voice signals, with an effective bandwidth of only 3100 Hz (300 to 3400), a 4 kHz bandwidth is adequate. The spectra of signals produced by modems for voice band transmission also fit well in this bandwidth. Another potential problem is **intermodulation noise**, on a long link, the nonlinear efects of amplifiers on a signal in one channel could produce frequency components in other channels.

3.3 TIME DIVISION MULTIPLEXING

A circuit switching technique in which time slots in a time multiplexed stream of data are manipulated to pass data from an input to an output *i.e.*, the division of a transmission facility into two or more channels by alloting the facility to several different information channels, one at a time.

Time division multiplexing (TDM) leaves the frequency spectrum intact but divides the transmission time into pieces, or slots, that are assigned periodically to the different devices sharing the link which is shown in figure 3.5.

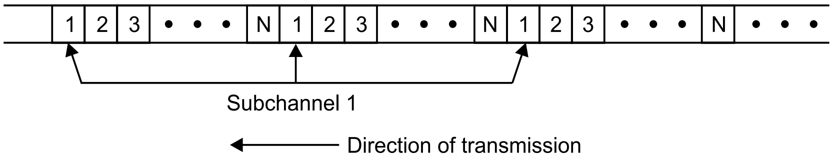


Fig. 3.5 Slots used for time division multiplexing

The multiplexer takes incoming traffic and delays it into its assigned time slot. This is commonly done one character or one bit at a time. The receiving demultiplexer must carefully clock incoming data to separate it into slots corresponding to the originals which is shown in figure 3.6.

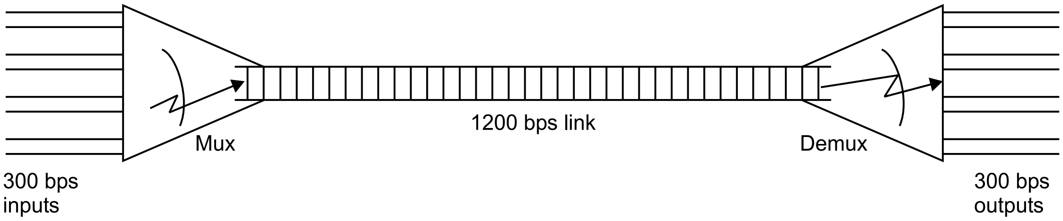


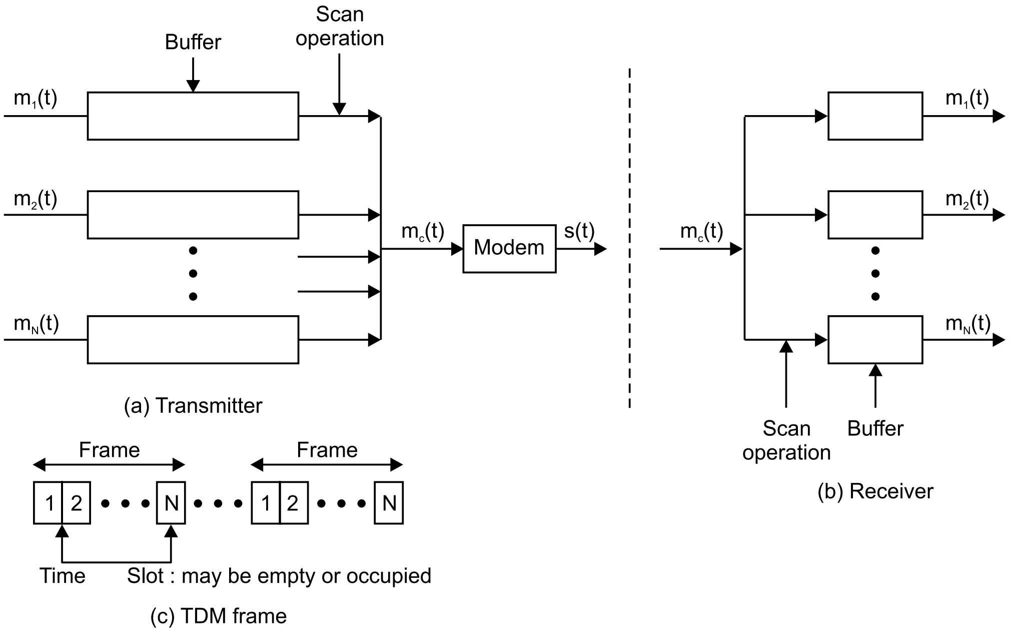
Fig. 3.6 Time division multiplexing

The effective transmission rate made available to each device through its sub-channel is the total speed of the link divided by the number of sub-channels *i.e.* by the number of devices that could share the link. When the allocation of slots to create sub-channels is a fixed assignment. The sharing scheme is called **synchronous** TDM commonly abbreviated as **STDM** because the time slots do not vary.

Synchronous time-division multiplexing is possible when the achievable data rate of the medium exceeds the data rate of digital signals to be transmitted. Multiple digital signals can be carried on a single transmission path by interleaving portions of each signal in time. The interleaving can be at the bit level or in blocks of bytes or larger quantities.

A generic depiction of a synchronous TDM system is provided in figure 3.7. A number of signals [$m_i(t)$, i = 1, N] are to be multiplexed onto the same transmission medium. The signals carry digital data and are generally digital signals. The incoming data from each source are briefly buffered. Each buffer is typically one bit or one character in length. The buffers are scanned sequentially to form a composite digital data stream $m_c(t)$. The scan operation is sufficiently rapid so that each buffer is emptied before more data can arrive. Thus, the data rate of $m_c(t)$ must at least equal the sum of the data rates of $m_i(t)$. The digital signal $m_c(t)$ may be transmitted directly or passed through a modem so that an analog signal is transmitted. In either case, transmission is typically synchronous.

The transmitted data may have a format something like figure 3.7(b). The data are organized into frames. Each frame contains a cycle of time slots. In each frame, one or more slots is dedicated to each data source. The sequence of slots dedicated to one source, from frame to frame is called a channel. The slot length equals the transmitter buffer length, typically a bit or a character.



(a) Transmitter

(b) Receiver

(c) TDM frame

Fig. 3.7 Synchronous TDM

The character-interleaving technique is used with asynchronous sources. Each time slot contain one character of data.

At the receiver, the interleaved data are demultiplexed and routed to the appropriate destination buffer. For each input source $m_i(t)$ there is an identical output source which will receive the input data at the same rate at which it was generated.

An outgrowth of synchronous TDM is called **Statistical TDM (STATDM)** uses dynamic reassignment of idle sub-channels to accommodate devices with data waiting to be sent. Only those devices with traffic share the entire capacity of the link eliminating waste as shown in figure. 3.8.

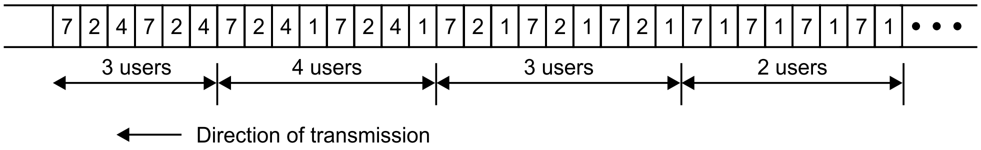


Fig. 3.8 Statistical TDM

If only one device makes a request, all the slots can be assigned to it and the transmission proceeds at the maximum rate the link provides. A static assignment could provide only the one low rate determined from how often the assigned slot occurs.

Statistical TDM is an improvement only in situations where the connected devices can take advantage of the extra link capacity available, when some devices are idle. A single terminal operating at 300 bps cannot benefit from assignment of all the slots on a 1200 bps link because it can only provide data to the link at the lower rate.

However more than four 300 bps terminals might share a single 1200 bps link if their patterns of usage were such that on the average, four or fewer have traffic to transmit simultaneously.

This ability to serve more devices than fixed assignment of the transmission capacity would allow is called concentration.

3.4 WAVELENGTH DIVISION MULTIPLEXING

For fiber optic channels, a variation of frequency division multiplexing is used.

Therefore, technically **optical FDM** is known as **Wavelength Division Multiplexing** *i.e.* (WDM) because humans see frequencies of visible light as colours, however engineers sometimes use the informal term **color division multiplexing** and joke about the carriers being “red”, “orange”, “blue” and so on.

Wavelength division multiplexing operates by sending multiple light waves across a single optical fiber. At the receiving end, an optical prism is used to separate the frequency, which is described in simple way.

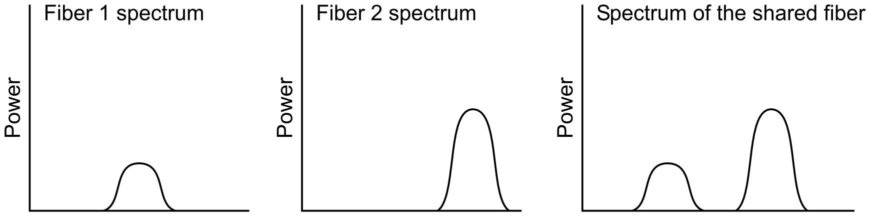


Fig. 3.9 (a)

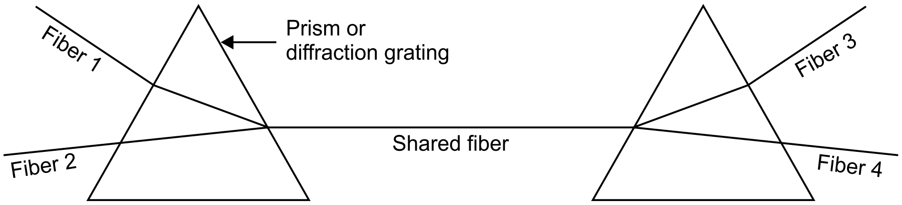


Fig. 3.9 (b) Wavelength division multiplexing

Here two fibers come at a prism, each with its energy in a different band. The two beams are passed through the prism or a diffraction grating and combined onto a single shared fiber for transmission to a distant destination, where they are split again.

As long as each channel has its own frequency range and all the ranges are disjoint, they can be multiplexed together on the long-haul fiber.

The only difference with electrical FDM is that an optical system using a diffraction grating is completely passive and thus highly reliable.

The reason for why WDM is popular? Is that the energy on a single fiber is typically only a few GHz. wide because it is currently impossible to convert between electrical and optical media any faster, since bandwidth of a single fiber band is about 25,000 GHz. There is great potential for multiplexing many channels together over long-haul routes. A necessary condition, however is that the incoming channels use different frequencies.

See figure 3.9(b), we have a fixed wavelength system. Bits from fiber 1 go to fiber 3 and bits from fiber 2 go to fiber 4. It is not possible to have bits go from fiber 1 to fiber 4.

However, it is also possible to build WDM systems that are switched. In such a device, there are many input fibers and many output fibers and the data from any input fiber can go to any output fiber.

Typically, the coupler is a passive star, with the light from every input fiber illuminating the star. Although spreading the energy over *n* outputs dilutes it by a factor *n*, such systems are practical for hundreds of channels.

QUESTIONNAIRES

1. What is multiplexing? Explain TDM in detail.
2. Discuss wavelength division multiplexing.
3. Differentiate FDM and TDM.

Chapter 4

INTRODUCTION TO COMPUTER NETWORK

INTRODUCTION

Each of the past three centuries has been dominated by a single technology. The eighteenth century was the time of the great mechanical systems accompanying the Industrial Revolution.

- The nineteenth century was the age of the steam engine.
- During the twentieth century, the key technology has been information gathering, processing and distribution.

These areas are rapidly converging, and the differences between collecting, transporting, storing and processing information are quickly disappearing. Organisations with hundreds of offices spread over a wide geographical area routinely expect to be able to examine the current status of even their most remote outpost at the push of a button. As our ability to gather, process and distribute information grows, the demand for even more sophisticated information processing grows even faster.

Although the computer industry is young compared to other industries *e.g.* automobiles and air transportation, computers have made spectacular progress in a short time. During the first two decades of their existence, computer systems were highly centralised, usually within a single large room. Not frequently, this room had glass walls, through which visitors could gawk at the great electronic wonder inside. A medium-size company or university might have had one or two computers, while large institutions had at most a few dozen. The idea that within 10 years equally powerful computers smaller than postage stamps would be mass produced by the millions was pure science fiction.

The merging of computers and communication has had a profound influence on the way computer systems are organised. The concept of the “computer center” as a room with a large computer to which users bring their work for processing is rapidly becoming obsolete. This model has not one, but at least two flaws : The concept of a single large computer doing all the work, and the idea of user bringing work to the computer instead of bringing the computer to the users.

The old model of a single computer serving all of the organisations computational needs, is rapidly being replaced by one in which a large number of separate, but interconnected computers do the job. These systems are called computer networks.

Therefore computer network means an interconnected collection of autonomous computer, “If one computer can forcibly start, stop or control another one, the computers are not autonomous”.

4.1 NEED OF COMPUTER NETWORKS

Computer Network satisfy a broad range of purposes and meet various requirements. Need of computer network arises for various purposes, and these are:

1. To provide sharing of resources such as information or processors.
2. To provide inter-process communication among users and processors.
3. To provide distribution of processing functions.
4. To provide centralised control for a geographically distributed system.
5. To provide centralised management and allocation of network resources.
6. To provide compatibility of dissimilar equipment and software.
7. To provide network users with maximum performance at minimum cost.
8. To provide an efficient means of transport large volumes of data among remote locations.

4.2 ADVANTAGES OF COMPUTER NETWORKS

These purposes must be fulfilled by various **advantages** of networks.

1. Resource Sharing

Resource sharing means the goal is to make all programs, data and equipment available to anyone on the network without regard to the physical location of the resource and the user.

Example:

Suppose a user happens to be 1000 km away from his data should not prevent him from using the data as though they were local. Also load sharing is another aspect of resource sharing.

2. High Reliability

Network provides high reliability by having alternative sources of supply.

Example:

Suppose all files could be replicated on two or three machines, so if one of them is unavailable (due to a hardware failure), the other copies could be used.

For military, banking, air traffic control, and man other applications, the ability to continue operating the face of hardware problems is of great importance.

3. Low Cost/Saving Money

Small computers have a much better price/performance ratio than large one. Mainframes are roughly a factor of fourty faster than the fastest single chip microprocessors, but they cost a thousand times more. This imbalance has caused many system designers to build systems consisting of powerful personal computers, as per user, with data kept on one or more shared file server machines.

4. Communications

Another goal of setting up a computer network has little to do with technology at all. A computer network can provide a powerful communication medium among widely separated people. Using a network, it is easy for two or more people who live far apart to write a report together. *i.e.* when one author makes a change to the document, which is kept online, the others can see the change immediately, instead of waiting several days for a letter.

4.3 USES OF COMPUTER NETWORKS

1. ***Access to remote programs:*** A company that has produced a model simulating the world economy may allow its clients to log in over the network and run the program to see how various projected inflation rates, interest rates, and currency fluctuations might affect their business. This approach is often preferable to selling the program outright, especially if the model is constantly being adjusted or requires an extremely large mainframe computer to run.
2. ***Access to remote data bases:*** It may soon be easy for the average person sitting at home to make reservations for aeroplanes, trains, buses, boats, hotels, restaurants, theatres and so on, anywhere in the world with instant confirmation.

   Home banking and the automated newspaper also fall in this category.
3. ***Value-added communication facilities:*** High-quality communication facilities tend to reduce the need for physical proximity. Everyone in the world, have an ability to send and receive electronic mail.

   These mails are also be able to contain digitized voice, still pictures and possibly even moving television and video images.
4. Using for entertainment purpose.
5. Accessing the information systems like world wide web, which contains almost any information.

4.4 NETWORK MODELS

1. *Centralised network model:* Here the terminals allows user has to enter data. But the processing is done on the server. It gives the ability to the user to access the data from the remote location.
2. *Distributed network model:* Here data storage and processing is done on the local computer. Hence the computers used in the distributed network are capable of working as stand alone. But can be network together to increase functionality.

4.5 CATEGORIES OF NETWORKS AND INTERNETWORKS

Today when we speak of networks, we are generally referring to three primary categories based on its size, its ownership, the distance it covers and its physical architecture.

- Local-Area Networks.
- Metropolitan-Area Networks.
- Wide-Area Networks.

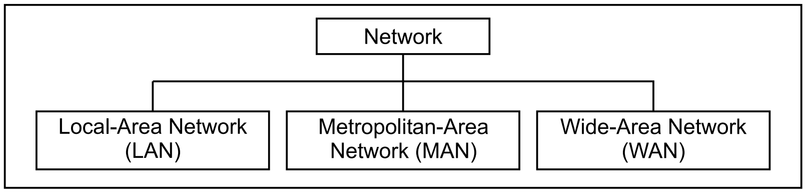
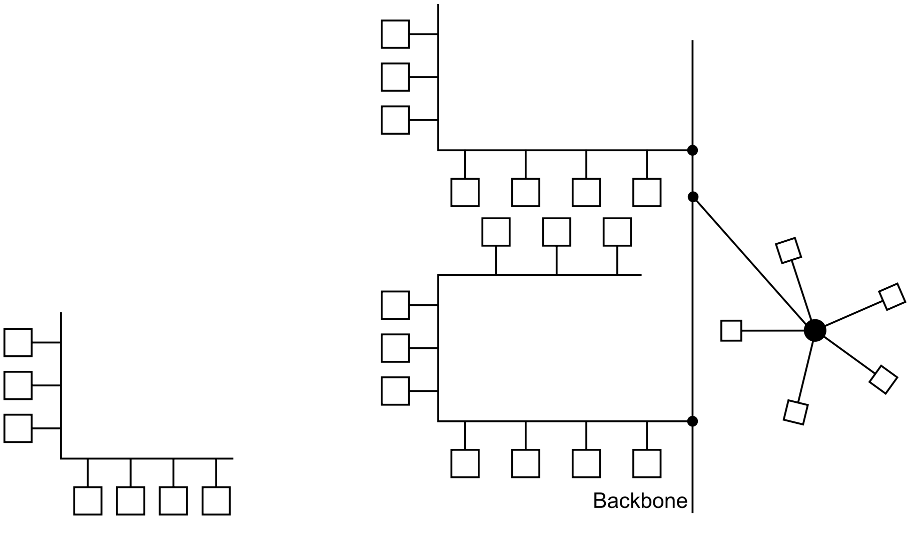


Fig. 4.1 Categories of networks.

Local-Area Network (LAN)

A Local-Area Network (LAN) is generally a privately owned network within a single office, building or campus covering a distance of a few kilometers shown in given Fig. 4.2.



(a) Single building LAN

(b) Multiple building LAN

Fig. 4.2 Local Area Network

- The main reason for designing a LAN is to share resources such as disks, printers, programs and data.
- It also enables the exchange of information.
- LAN having data rate of 4 Mbps to hundreds of Mbps.
- LANs typically can use the star, bus or a ring topology.
- Example Ethernet LANs, Token Bus LANs, Token Ring LANs, FDDI.

Metropolitan–Area Network (MAN)

- A Metropolitan-Area Network (MAN) is designed to cover an entire city.
- It may be a single network such as a cable television network, or it may be a means of connecting a number of LANs into a larger network so that resources may be shared LAN-to-LAN as well as device-to-device.
- A Metropolitan-Area Network is shown in Fig. 4.3.
- A MAN may be wholly owned and operated by a private company, or it may be a service provided by a public company, such as local telephone company.
- Many telephone companies provide a popular MAN device called switched Multimegait Data Services.
- A MAN has a larger geographical scope compared to a LAN and can range from 10 km to a few hundreds km in length.
- A typical LAN operates at a speed of 1.5 to 150 Mbps.

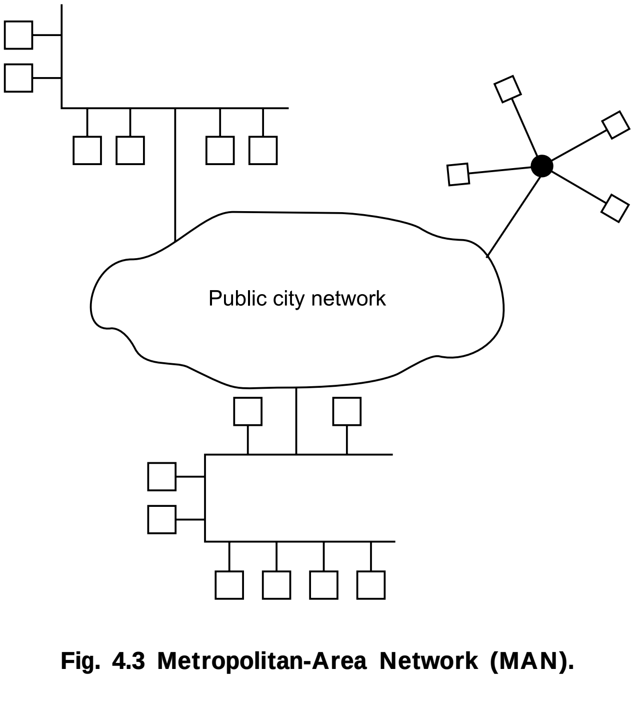


Fig. 4.3 Metropolitan-Area Network (MAN).

Wide-Area Network (WAN)

- A WAN is designed to interconnect computer systems over large geographic scope, such as country, a continent, or even the whole world, as shown in figure 4.4.

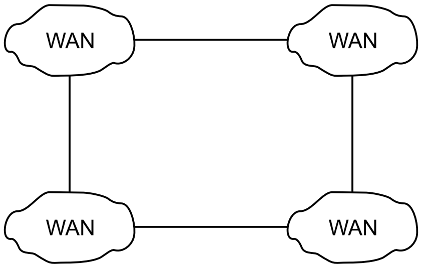


Fig. 4.4 Wide-Area Network (WAN).

- A WAN speed ranges from 1.5 Mbps to 100 Gbps.
- WANs may utilize public, leased or private communication devices, usually in combinations, and can therefore span an unlimited number of miles.
- A good example of such a network is internet, which has a connection to similar networks in other countries.

Internetworks

- When two or more networks are connected, they become an internetwork, or internet as shown in Fig. 4.5.

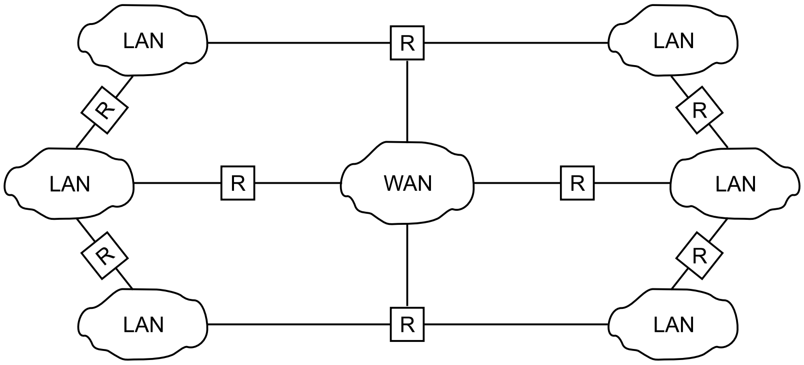


Fig. 4.5 Internetwork (internet)

- The boxes labeled R represent routers.
- Individual networks are joined into internetworks by the use of internetworking devices.
- These devices, which include routers and gateways.
- The term internet (lower case *i*) should not be confused with the internet (upper case I). The first is a generic term used to mean an interconnection of networks. The second is the name of a specific worldwide network.

4.6 LINE CONFIGURATION

Line configuration means the way two or more communication devices attach to a link.

A Link is the physical communication pathway that transfers data from one device to another.

For communication to occur, two devices must be connected in same way to the same link at the same time.

There are two possible line configurations.

1. Point-to-Point.
2. Multipoint.

1. Point-to-Point

- A point-to-point line configuration provides a dedicated link between two devices.
- The entire capacity of the channel is reserved for transmission between those two devices.
- Most point-to-point line configurations use an actual length of wire or cable to connect the two ends, but other options, such as microwave or satellite links, are also possible as shown in Fig. 4.6.

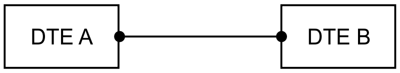


Fig. 4.6 Point-to-Point line configuration.

When you change television channels by infrared remote control, you are establishing a point-to-point line configuration between the remote control and the television's control system.

2. Multipoint

- A multipoint (also called multidrop) line configuration is one in which more than two specific devices share a single link as shown in Fig. 4.7.

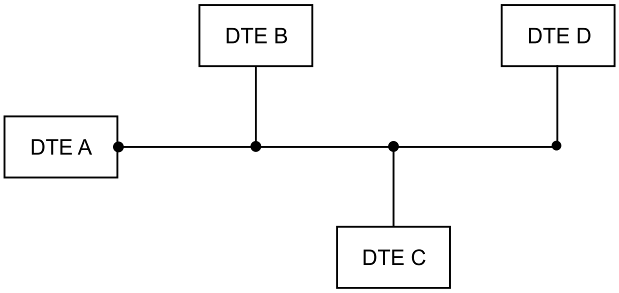


Fig. 4.7 Multipoint line configuration.

- In a multipoint environment, the capacity of the channel is shared, either spatially or temporaily. If several devices can use the link simultaneously, it is a spatially shared line configuration. If users must take turns, it is a time-shared line configuration.

4.7 NETWORK TOPOLOGIES

The term "**TOPOLOGY**" refers to the way in which the end points or stations/computer systems, attached to the networks, are interconnected.

We have seen that a topology is essentially a stable geometric arrangement of computers in a network. If you want to select a topology for doing networking. You have attention to the following points.

- Application S/W and protocols.
- Types of data communicating devices.
- Geographic scope of the network.
- Cost.
- Reliability.

Depending on the requirement there are different Topologies to construct a network.

(1) Mesh topology.

(2) Star topology.

(3) Tree topology.

(4) Bus topology.

(5) Ring topology.

(6) Cellular topology.

- Ring and mesh topologies are felt convenient for peer to peer transmission.
- Star and tree are more convenient for client server.
- Bus topology is equally convenient for either of them.

Mesh Topology

In mesh topology each and every computer is connected to each other with direct point to point link.

A fully connected mesh network therefore has n (n–1/2) physical channels to link n devices.

To accommodate these, every device on the network must have n–1 input/output parts.

Advantages

- Use of dedicated links eliminates the traffic problems.

- It is robust, *i.e.* if one link becomes unusable it does not incapacitate the entire system.
- Privacy is maintained since the message travels along the dedicated lines.
- Point-to-point link makes fault identification and fault isolation easy.

Disadvantages

- The amount of cabling required is high.
- The number if I/O ports required is high.

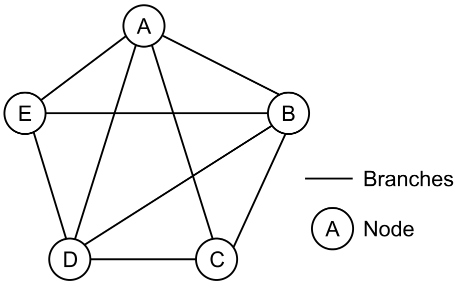


Fig. 4.8 Mesh topology

Star Topology

In a star topology, cables run from every computer to a centrally located device called a **HUB**.

Star topology networks require a central point of connection between media segment. These central points are referred to as Hubs.

Hubs are special repeaters that overcome the electromechanical limitations of a media.

Each computer on a star network communicates with a central hub that resends the message either to all the computers. (In a broadcast network) or only the destination computer. (In a switched network).

Ethernet 10 base T is a popular network based on the star topology.

Advantages

- Easy to modify and to add new computers without disturbing the rest of the network.
- Less expensive than mesh topology.
- Each device needs only one link and one port.
- Easy to install and configure.
- Easy to diagnose network faults.

- Single computer failure does not affect the network.
- Ordinary telephone cables can be used.

Disadvantages

- More cabling is required as compare to others.
- Failure of the central hub brings the entire network down.

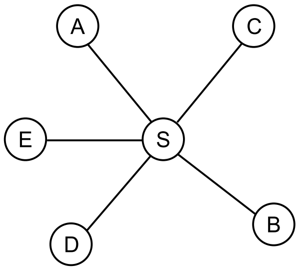


Fig. 4.9 Star topology

Bus Topology

- A bus topology is multipoint.
- One long cable acts as a backbone to link all the devices in the network.
- Nodes are connected to the bus cable by drop lines and taps.
- A drop line is a line running between the device and main cable.
- A tap is connector that splices into the main cable. There is a limit on the number of taps used and distance between the taps.

Advantages

- Simple, reliable and easy to use.
- Easy to installation and cheaper than when compared with others.
- Less cabling.

Disadvantages

- Can be used in relatively small networks.
- All computers share the same bus.
- Reconfiguration is difficult.
- Fault identifications is difficult.
- Adding new nodes is difficult.
- A fault on the cable stops all transmission.

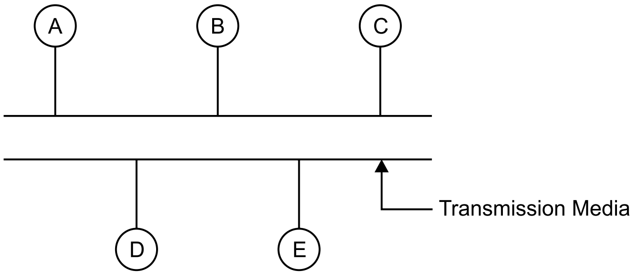


Fig. 4.10 Bus topology

Ring Topology

In ring topology, each device has a dedicated point-to-point line configuration only with two devices on either side of it.

A signal is passed along the ring in one direction, from device to device until it reaches its destination.

Each device in the ring has a repeater. When the devices receive the signal intended for the other node, it just regenerates the bits and passes them along.

Ring network passes a **token**.

A token is a short message with the electronic address of the receiver.

Each network interface card is given a unique electronic address, which is used to identify the computer on the network.

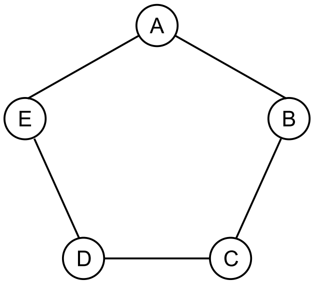


Fig. 4.11 Ring topology

Advantages

- Easy to install and reconfigure.
- Adding/deleting the new device is easy as only two connections have to be adjusted.
- Fault isolation is simplified.
- No terminators required.

Disadvantages

- A break in the ring can stop the transmission the entire network.
- Difficult to troubleshoot.
- Adding/removing computer distrupts the entire network.
- Expensive when compared with other topologies.
- When one computer fails overall network distrupts.

Tree Topology

It is similar to the star network, but the nodes are connected to the **secondary hub** that in turn is connected to the central hub.

The central hub is the active hub.

The active hub contains the repeater, which regenerates the bits pattern it receives before sending them out.

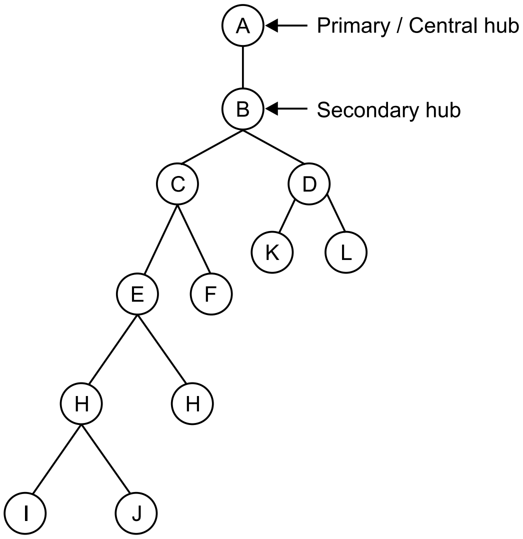


Fig. 4.12 Tree topology

The secondary hub can be either active or passive.

A passive hub provides a simple physical connection between the attached devices.

Advantages and **Disadvantages** of the tree are same as that of the star network. Also, the addition of the secondary hub allows more devices to be attached to the central hub. It also allow the network to isolate priorities communication from different computers.

The tree topology structure is shown in the figure 4.12.

Cellular Topology

The cellular topology is applicable only in case of wireless media that does not require cable connection.

In wireless media, each point transmits in a certain geographical area called a **cell**.

Each cell represents a portion of the total network area.

Devices that are in the cell communicate through a central hub. Hubs in different cells are interconnected. They route data across the network and provide a complete network infrastructure.

The data is transmitted in the **cellular digital packet data** (**CDPD**) format.

Advantages

- Troubleshooting is easy.
- Hub-to-hub fault tracking is more complicated, but allows simple fault isolation.

Disadvantages

- When the central hub fails, all the units in the assigned range of cell are affected.

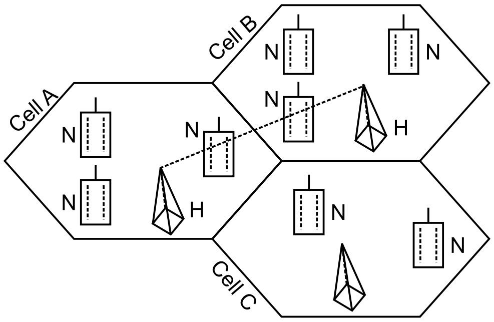


Fig. 4.13 Cellular topology

4.8 STUDY OF REFERENCE MODELS

In this section we will discuss important network architectures, mean different network reference models.

4.8.1 Protocol Hierarchies

To reduce their design complexity, most networks are organised as a series of layers or levels, each one built upon its predecessor. The number of layers, the name of each layer, the contents of each layer, and the function of each layer differ from network to network. However, in all networks, the purpose of each layer is to offer certain services to the higher layers, shielding those layers from the details of how the offered services are actually implemented.

Layers N on one machine carries on a conversation with layer N on another machine. The rules and conventions used in this conversation are collectively known as the layer N protocol. Basically, **a protocol is an agreement between the communicating parties on how communication is to proceed** (For eg. a seven layer network). The entities comprising the corresponding layers on different machines are called **peer processes**.

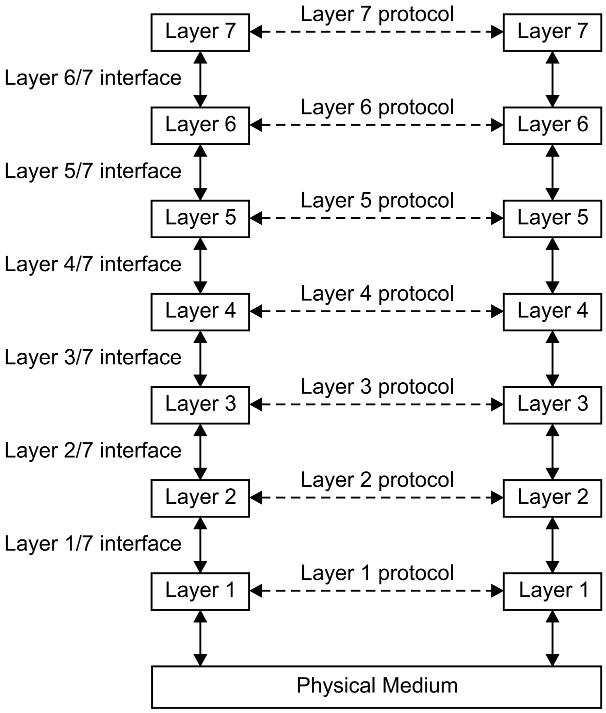


Fig. 4.14 Layers, Protocols and Interfaces

In reality, no data are directly transferred from layer N on one machine to layer N on another machine. Instead, each layer passes data and control information to the layer immediately below it, until the lowest layer is reached. Below layer 1 is the physical medium through which actual communication occurs. Between each pair of adjacent layers there is an interface. The interface defines which primitive operations and services the lower layer offers to the upper one. When network designers decide how many layers to include in a network.

What each one should do, one of the most important considerations is defining clean interfaces between the layers. Doing so in turn, requires that each layer performs a specific collection of well understood functions. In addition to minimising the amount of information that must be passed between layers, clean cut interfaces also make it simpler to replace the implementation of one layer with a completely different implementation because all that is required of the new implementation is that it offer exactly the same set of services to its upstairs neighbour as the old implementation did.

The set of layers and protocols is called the **network architecture.** The specification of the architecture must contain enough information to allow an implemented to write the program or build the hardware for each layer so that it will correctly obey the appropriate protocol.

One example is given, to explain the idea of multilayer communication. Imagine two philosophers, one in Kenya and one in Indonesia, who want to communicate, since they have no common language, they each engage a translator, each of whom in turn contacts an engineer. Philosopher 1 wishes to convey his affection for oryctolagus cuniculus to his peer. To do so, he passes a message across the 2/3 interface to his translator, who might render it, as "I like rabbits" or "Jaime des lapins or" "IK hou van kon" "nen" depending on the layer 2 protocol.

The translator then gives the, message to his engineer for transmission by telegram, telephone, computer network, or some other means depending on what the two engineers have agreed. On in advance (the layer 1 protocol). When the message arrives, it is translated into Indonesian and passed across the 2/3 interface to philosopher 2. Note that each protocol is completely independent of the other ones as long as the interfaces are not changed. The translators can switch from French to Hindi at will, provided that they both agree, and neither changes his interface with either layer 1 or layer 3. Now consider a more technical example ; how to provide communication to the top layer of the, seven layer network. A message, M, is produced by a process running in layer 7. The message is passed from layer 7 to layer 6 according to the definition of the layer 6/7, interface, layer 6 transforms the message in certain ways and then passes the new message M to layer 5 across the layer 5/6 interface. Layer 5, does not modify the message but simply regulates the direction of flows.

In many networks, there is no limit to the size of message accepted by layer 4, but there is a limit imposed by layer 3. Consequently, layer 4 must break up the incoming messages into smaller units, prepending a header to each unit. In many layer, headers also contain sizes, times and other control fields.

Layer 3 decides which of the outgoing lines to use, attaches its own headers and passes the data to layer 2. Layer 2 adds not only a header to each piece, but also a trailer, and gives the resulting unit to layer 1 for physical transmission. At the receiving machine the message moves.

4.8.2 Design Issues for the Layers

Some of the key design issues that occur in computer networking are present in several layers. Below, we will briefly mention some of the more important ones.

- Every layer must have a mechanism for connection establishment.
- Another set of design decision concerns the rules for data transfer. In some systems, data only travel in one direction (simplex communication). In other they can travel in either direction, but not simultaneously (half duplex communication). In still others they travel in both directions at once (full duplex communication).
- Not all communication channels preserve the order of messages sent on them. To deal with a possible loss of sequencing, the protocol must make explicit provision for the receiver to allow the pieces to be put back together properly.
- An issue that occurs at every level is how to keep a fast sender from swamping a slow receiver with data.

- Another problem that must be solved at several levels is the inability of all processes to accept arbitrarily long messages. This property leads to mechanisms for disassembling, transmitting, and then reassembling messages.
- When it is inconvenient or expensive to set up a separate connection for each pair of communicating processes, the underlying layer, may decide to use the same connection for multiple, unrelated conversations. As long as this multiplexing and demultiplexing is done transparently, it can be used by any layer. Multiplexing is needed in the physical layer.
- When there are multiple paths between source and destination, a route must be chosen. Sometimes this decision must be split over two or more layers.

4.8.3 The OSI Reference Model

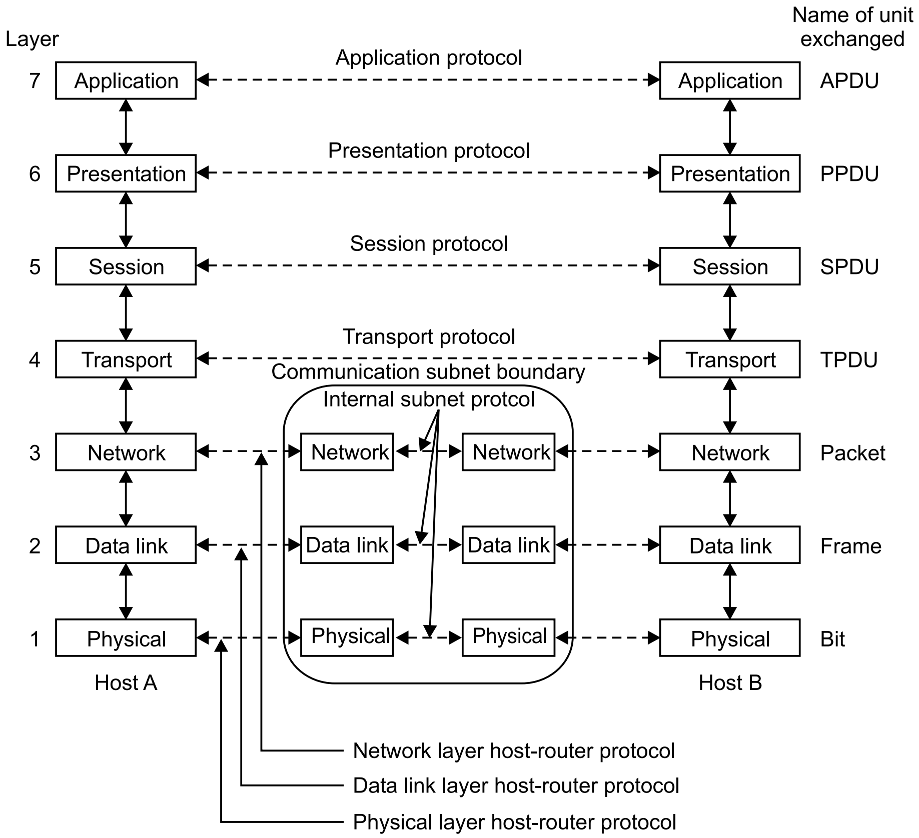


Fig. 4.15 The OSI reference model

The OSI model is shown in the figure 4.15. This model is based on a proposal developed by the **International Standards Organization (ISO)** as a first step towards international standardization of the protocols used in the various layers, (Day & Zimmerman, 1983). The model is called the **ISO OSI (Open Systems Interconnection)** Reference Model because it deals with connecting open systems, *i.e.*, the systems that are open for communication with other systems.

The OSI Model has seven layers. The **principle** that were applied to arrive at the seven layers are as follows.

1. A layer should be created where a different level of abstraction is needed.
2. Each layer should perform a well defined function.
3. The function of each layer should be chosen with an eye toward defining internationally standardized protocols.
4. The layer boundaries should be chosen to minimize the information flow across the interfaces.
5. The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity, and small enough that the architecture does not become unwidely.

Now we will discuss each layer of the model, starting at the bottom layer. Note that the OSI model itself is not a network architecture because it does not specify the exact services and protocols to be used in each layer. It just tells what each layer should do.

Physical layer

The physical layer is concerned with transmitting raw bits over a communication channel. The design issues have to do with making sure that when one side sends a 1 bit, it is received by other side as a 1 bit, not as a 0 bit. Typical questions here are how many volts should be used to represent a 1 and how many for a 0? How many microseconds a bit lasts? Whether transmission may proceed simultaneously in both direction, how the initial connection is established and how it is torn down when both sides are finished and how many pins the network connector has and what each pin is used for.

The design issues here largely deal with mechanical, electrical and procedural interfaces, and the physical transmission medium, which lies below the physical layer.

Data link layer

The main task of the data link layer is to take a raw transmission facility and transform it into a line that appears free of undetected transmission errors to the network layer. It accomplishes this task by having the sender break, the input data up into data frames.

Transmit the frames sequentially and process the acknowledgement frames sent back by the receiver. Since the physical layer merely accepts and transmits a stream of bits without any regard to structure, it is up to the data link layer to create and recognize frame boundaries. This can be accomplished by attaching special bit patterns to the beginning and end of the frame.

The data link layer may offer several different service classes to the network layer each of a different quality and with a different price.

Another issue that arises in the data link layer is how to keep a fast transmitter from drawing a slow receiver in data.

Broadcast network have an additional issue in the data link layer, how to control access to the shared channel.

Network layer

The network layer is concerned with controlling the operation of the subnet. A key design issue is determining how packets are routed from source to destination. Routes can be based on static tables that are "wired into the network and rarely changed." They can be highly dynamic, being determined a new for each packet, to reflect the current network load.

If two many packets are present in the subnet at the same time, they will get in each other's way, forming bottlenecks. The control of such congestion also belongs to the network layer.

When a packet has to travel from one network to another to get to its destination, many problems can arise. The addressing used by second network may be different from first one. The second one may not accept the packet at all because it is too large. The protocols may differ and so on. It is up to the network layer to overcome all these problems to allow heterogeneous network to be interconnected.

In broadcast networks, the routing problem is simple, so the network layer is often thin or even non-existent.

Transport layer

The basic function of the transport layer is to accept data from the session layer, split it up into smaller units if needed, pass these to the network layer and ensure that the pieces all arrive correctly at the other end.

The transport layer is required to make the multiplexing transparent to the session layer.

The transport layer also determines what type of service to provide the session layer and ultimately the users of the network. The most popular type of transport connection is an error-free point-to-point channel, that delivers messages in the order in which they were sent.

The transport layer is a true end-to-end layer, from source to destination.

To multiplex several message streams onto one channel, the transport layer must take care of establishing and deleting connections across the network. There must be also a mechanism to regulate the flow of information, so that a fast host cannot overrun a slow one. Such a mechanism is called **flow control**.

Session layer

The session layer allows users on different machines to establish sessions between them. A session allows ordinary data transport.

A session might be used to allow a user to log onto a remote timesharing system or to transfer a file between two machines.

One of the service of the session layer is to manage dialogue control.

The session can allow traffic to go in both directions at the same time or in only one direction at a time.

Another service of session layer is "**Token Management**". [For some protocols, it is essential that both sides do not attempt the same operation at the same time. To manage these activities, the session layer provides token that can be exchanged.] Only the side holding the token may perform the critical operation.

Also another session service is "**synchronization**". *i.e.* consider the problem that occur when trying to do a 2-hour file transfer between two machines with a 1-hour mean time between crashes. After each transfer was aborted, the whole transfer would have to start over again and would probably fail again the next time as well.

To eliminate this problem, the session layer provides a way to insert checkpoints into the data stream, so that after a crash, only the data transferred, after the last checkpoint have to be repeated.

Presentation layer

The presentation layer performs certain function that are requested sufficiently often to warrant finding a general solution for them, rather than letting each user solve the problems.

The presentation layer is concerned with the syntax and semantics of the information transmitted.

The presentation layer is used as an encoding data in a standard agreed upon way. The presentation layer manage these abstract data structures and converts from the representation used inside the computer to the network standard representation and back.

Application layer

The application layer contains a variety of protocols that are commonly needed. Suppose there are number of terminal types in the world, and the plight of a full screen editor that is supposed to work over a network with many different terminal types, each with different screen layouts, escape sequences for inserting and deleting text, moving the cursor, etc.

To solve this problem we use to define an abstract nework virtual terminal that editors and other programs can be written to deal with.

Another application layer function is file transfer, different file systems have different file naming conventions, different ways of representing text lines and so on. Transferring a file between two different systems requires handling these and other incompatibilities.

4.8.4 The TCP/IP Reference Model

Let us now turn from the OSI reference model to the reference model used in the grandparent of all computer networks, the ARPANET, and its successor, the worldwide internet although we will give a brief history of the ARPANET later, it is useful to mention a few key aspects of it now. The ARPANET was a research network sponsored by the DOD

(U.S. Department of Defense). It eventually connected hundreds of universities and government installations using leased telephone lines. When satellite and radio networks were added later, the existing protocols had trouble interworking with term, so a new reference architecture way needed. Thus the ability to connect multiple networks together in *so* seamless way was one of the major design goals from the very beginning. This architecture later became known as the TCP/IP Reference Model.

Given the DOD's worry that some of its precious hosts, routers and internet-work gateways might get blown to pieces at a moment notice, another major goal was that the network be able to survive loss of subnet hardware, with existing conversations not being broken off.

Host-to-Network layer

This is the bottom layer of TCP/IP referenced model. The TCP/IP reference model does not really say much about what happens here except to point out that the host has to connect to the network using some of the protocols so it can send IP packets over remove it. This protocol is not defined and varies the from host-to-host and network-to-network.

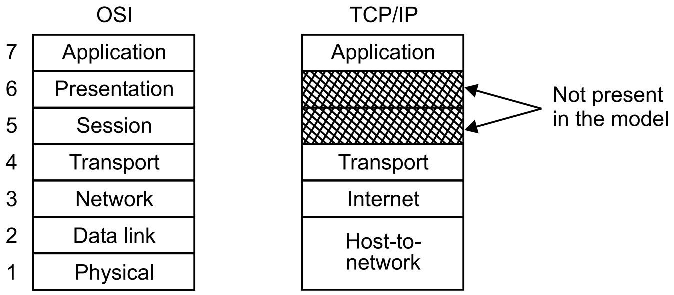


Fig. 4.16 The TCP/IP reference model.

Internet layer

The internet layer is the linchpin that holds the whole architecture together. Its function is to permit hosts to injects packets into any network, and have them travel independently to the destination (i.e., on different network). They may even arrive in a different order than they were sent, in which case it is the job of higher layers to rearrange them. If in-order delivery is desired.

The internet layer is used in mail system. The internet layer defines an official packet format and protocol called IP (Internet Protocol). The function of internet layer is to deliver IP packets where they are supposed to go. Packet routing is clearly the major issue here, as is avoiding congestion.

Transport layer

The layer above the internet layer in the TCP/IP model is now usually called the transport layer. It is designed to allow peer entities on the source and destination hosts to carry on a conversation. The same as in the OSI transport layer. Two end-to-end protocols have been defined here. The first one, **TCP (Transmission Control Protocol)** is a reliable connection oriented protocol that allows a byte stream originating on one machine to be delivered without error on any other machine in the internet. It fragment byte stream into discrete message and passes each one onto the internet layer. At the destination, the receiving TCP process reassembles the received messages into the output stream. TCP also handles flow control to make sure a fast sender cannot swamp a slow receiver with more messages than it can handle.

The second protocol in this layer, **UDP (User Datagram Protocol),** is an unreliable, connectionless protocol for applications that do not want TCP's sequencing or flow control and wish to provide their own. It is also widely used for one-shot, client-server type request-reply queries and applications in which prompt delivery is more important than accurate delivery, such as transmitting speech or video.

The relation of IP, TCP and UDP is shown in figure 4.17. Since the model was developed, IP has been implemented on many other network.

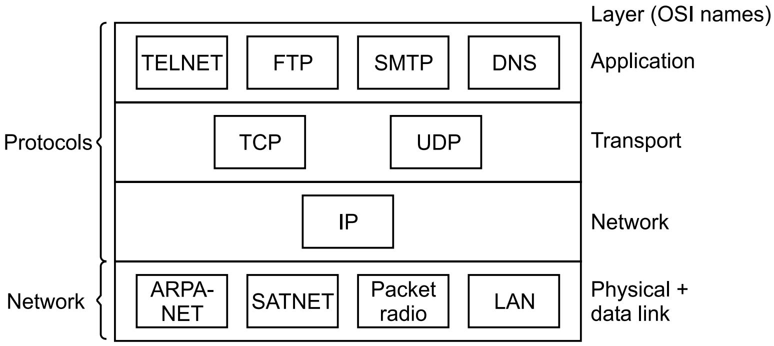


Fig. 4.17 Protocols and networks in the TCP/IP model initially.

Application layer

The TCP/IP model does not have session and presentation layers. No need for them was perceived, so they were not included. Experience with the OSI model has proven this view correct: they are of little use to most applications.

On top of the transport layer is the application layer. It contains all the higher-level protocols. The early ones included virtual terminal (TELNET), File transfer (FTP), and electronic mail (SMTP), as shown in figure 4.17.

The **TELNET** allows users on one machine to log into a distant machine and work there.

The **FTP** provides a way to move data efficiently from one machine to another.

The **SMTP** was originally just a kind of file transfer. Many other protocols have been added to these over the years, such as (**DNS**) the **Domain Name Service** for mapping host names onto their network addresses, **NNTP**, the protocol used for moving news articles around, and **HTTP**, the protocol used for fetching pages on the World Wide Web, and many others.

4.8.5 A Comparison of the OSI and TCP Reference Models

Similarities

- These two, layers are based on the concept of a stack of independent protocols.
- The functionality of the layers is roughly similar.

Examples

- In both models the layers up through and including the transport layer are there to provide an end-to-end network independent transport services to processes wishing to communicate. These layers form the transport provider.
- In both models, the layers above transport are application oriented users of the transport service.

Despite these fundamental similarities, the two models also have many differences. In this section we will focus on the key differences between the two reference models. It is important to note that we are comparing the reference models here, not the corresponding protocol stacks. The protocols themselves will be discussed later.

Three concept are central to the OSI model.

1. Services
2. Interfaces
3. Protocols

The OSI model originally clearly distinguishes between service, interface, and protocol.

The **service** definition tells what the layer does, not how entities above it access it or how the layer works. A layer's interface tells the processes above it how to access it. It specifies what the parameters are and what results to expect. The peer **protocols** used in a layer are the layer's own business.

The TCP/IP model did not originally clearly distingish between service, interface and protocol, although people have tried to retrofit it after the fact to make it more OSI-like.

As a consequence, the protocols in the OSI model are better hidden than in the TCP/IP model and can be replaced relatively easily as the technology changes.

The OSI reference model was devised before the protocols were invented. This ordering means that the model was not biased toward one particular set of protocols, which made it quite general.

With the TCP/IP the reverse was true : the protocols came first, and the model was really just a description of the existing protocols. There was no problem with the protocols fitting the model. They fit perfectly. The only trouble was that the model did not fit any other protocol stacks.

The OSI model has seven layers and the TCP/IP has four layers. Both have network, transport and application layers, but the other layers are different.

Another difference is in the area of connectionless versus connection oriented communication. The OSI model support both **connectionless and connection oriented communication** the network layer, but only connection oriented communication in the transport layer, where it counts. The TCP/IP model has only one mode in the network layer. (**Connectionless**) but supports both modes in the transport layer, giving the users a choice. This choice especially important for simple request-response protocols.

4.8.6 ATM

Asynchronous transfer mode (ATM), also known as cell relay, is in some ways similar to packet switching using X.25 and frame relay. Like packet switching and frame relay, ATM involves the transfer of data in discrete chunks. Also, like packet switching and frame relay, ATM allows multiple logical connections to be multiplexed over a single physical interface. In the case of ATM, the information flow on each logical connection is organized into fixed-size packets, called cells.

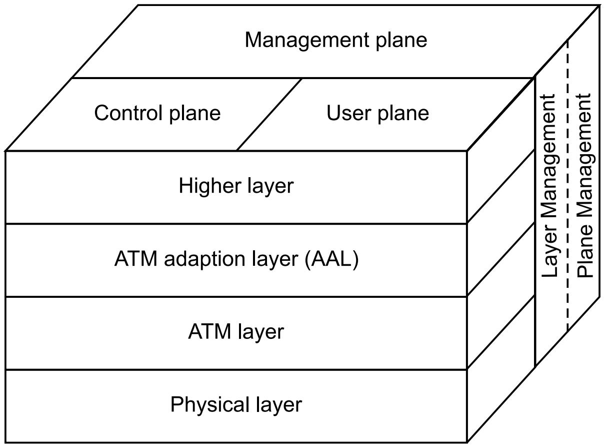


Fig. 4.18 ATM Protocol reference model.

ATM is a streamlined protocol with minimal error and flow control capabilities, this reduces the overhead of processing ATM cells and reduces the number of overhead bits required with each cell, thus enabling ATM to operate at high data rates. Further, the use of fixed-size cells simplifies the processing required at each ATM node, again supporting the use of ATM at high data rates.

The standards issued for ATM by ITU-T are based on the protocol architecture shown in the figure 4.18, which illustrate the basic architecture for an interface between user and network. The physical layer involves the specification of a transmission medium and a signal encoding scheme. The data rates specified at the physical layer include 155.52 Mbps and 622.08 Mbps. Other data rates, both higher and lower, are possible.

Two layers of the protocol architecture related to ATM functions. There is an ATM layer common to all services that provides packet transfer capabilities, and an ATM adaptation layer (AAL) that is service dependent. The ATM layer defines the transmission of data in fixed-size cells and also defines the use of logical connections. The use of ATM creates the need for an adaptation layer to support information transfer protocols not based on ATM. The AAL maps higher-layer information into ATM cells to be transported over an ATM network, then collects information from ATM cells for delivery to higher layers.

The protocol reference model makes reference to three separate planes.

- **User Plane:** Provides for user information transfer along with associated controls. (*e.g.* flow control, error control).
- **Control Plane :** Performs call control and connection control functions.
- **Management Plane :** Includes plane management, which performs management functions related to a system as a whole and provides coordination between all the planes, and layer management, which perform management functions relating to resources and parameters residing in its protocol entities.

4.9 NETWORK EXAMPLES

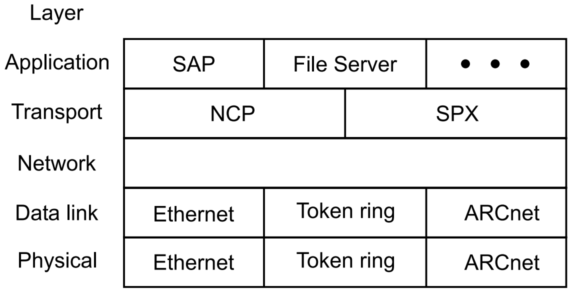
Numerous networks are currently operating around the world. Some of these are public networks run by common carriers or PTTs, others are research networks, yet others are co-operative networks run by their users, and still others are commercial or corporate networks. In the following sections we will take a look at a few current and historical networks to get an idea of what they are (or were) like and how they differ from one another.

In the following sections we will look at a few examples. These are the popular commercial LAN networking package, Novell Netware®, the Worldwide Internet (including its predecessors, the ARPANET and NSFNET), and the first gigabit networks.

NOVELL NETWARE

The most popular network system in the PC world is **Novel Netware**. It was designed to be used by companies downsizing from a mainframe to a network of PCs. In such systems, each user has a desktop PC functioning as a client. In addition, some number of powerful PCs operate as server providing file services, database services, and other services to a collection of clients. In other words, Novell Netware is based on the client-server model.

Netware uses a proprietary protocol stack illustrated in figure 4.19. It is based on the old Xerox Network System, XNS^{TM} but with various modifications. Novell Netware predates OSI and is not based on it. If anything, it looks more like TCP/IP than like OSI.



Layer			
Application	SAP	File Server	• • •
Transport	NCP	SPX	
Network			
Data link	Ethernet	Token ring	ARCnet
Physical	Ethernet	Token ring	ARCnet

Fig. 4.19 The Novell Netware reference model.

The physical and data link layers can be chosen from among various industry standards, including **Ethernet, IBM token ring,** and **ARCnet**. The network layer runs an unreliable connectionless internetwork protocol called **IPX**. It passes packet transparently from source to destination, even if the source and destination are on different networks. IPX is functionally similar to IP, except that it uses 10-byte addresses instead of 4-byte addresses.

Above IPX comes a connection-oriented transport protocol called **NCP** (Network Core Protocol). NCP also provides various other services besides user data transport and is really the heart of Netware. A second protocol, **SPX** is also available, but provides only transport. **TCP** is another option. Applications can choose any of them. The file system uses NCP and Lotus Notes® uses SPX, for example. The session and presentation layer do not exist. Various application protocols are present in the application layer.

As in TCP/IP, the key to the entire architecture is the internet datagram packet on top of which everything else is built. The format of an IPX packet is shown in figure 4.20. The **checksum field** is rarely used. Since the underlying data link layer also provides a checksum. The **packet length** field tells how long the entire packet is, header plus data. The **Transport Control** field counts how many networks the packet has transferred. When this exceeds a maximum, the packet is discarded. The **Packet type** field is used to mark various control packets. The two addresses each contain a 32-bit network number, a 48-bit machine number (the 802 LAN address), and 16-bit local address (socket) on that machine. Finally, we have the data, which occupy the rest of the packet, with the maximum size being determined by the underlying network.

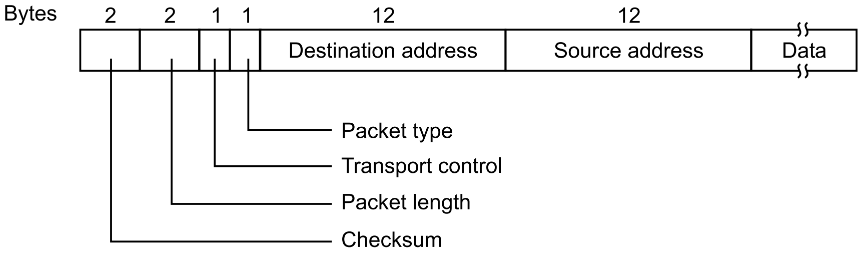


Fig. 4.20 A Novell Netware IPX packet.

About once a minute, each server broadcasts a packet giving its address and telling what services it offers. These broadcasts use, the **SAP (Service Advertising Protocol)** Protocol. The packets are seen and collected by special agent processes running on the router machines. The agents use the information contained in them to construct database of which servers are running where.

When a client machine is booted, it broadcasts a request asking where the nearest server is. The agent on the local router machine sees this request, looks in its database of servers and matches up the request with the best server. The choice of server to use is then sent back to the client. The client can now establish on NCP connection with the server. Using this connection, the client and server negotiate the maximum packet size. From this point on, the client can access the file system and other services using this connection. It can also query the server's database to look for other (more distant) servers.

The ARPANET

Let us now switch gears from LANs and WANs. In the mid-1960s, at the height of the cold war, the DOD wanted and control network that could service a nuclear war. Traditional Circuit-Switches telephone networks were considered too vulnerable since the loss of one line or switch would certainly terminate all conversations using them and might even partition the network. To solve this problem, DOD turned to its research arm, ARPA (later DARPA, now ARPA again), the (Periodically Defense) Advanced Research Project Agency.

ARPA was created in response to the Soviet Union's launching Sputnik in 1957 and had the mission of advancing technology that might be useful to the military. ARPA had no scientists or laboratories, in fact, it had nothing more than an office and a small (by Pentagon standard) budget. It did its work by issuing grants and contracts to universities and companies whose ideas looked promising to it.

Several early grants went to universities for investigating the then-radical idea of packet switching, something that had been suggested by Paul Baran in a series of RAND Corporation reports published in the early 1960s. After some discussions with various experts, ARPA decided that the network the DOD needed should be a packet-switched network, consisting of a subnet and host computers.

The subnet would consists of minicomputers called **IMPs (Interface Message Processors)** connected by transmission lines. For high reliability, each IMP would be connected to at least two other IMP. The subnet was to be a datagram subnet, so if some lines and IMPs were destroyed, messages could be automatically rerouted along alternative paths.

Each node of the network was to consist of an IMP and a host, in the same room, connected by a short wire. A host could send message of up to 8063 bits to its IMP, which would then break these up into packets of at most 1008 bits and forward them independently towards the destination. Each packet was the first electronic store-and-forward packet-switching network. (received in its entirety before being fowarded, so the subject was).

ARPA then put out a tender for building the subnet. Twelve companies bid for it. After evaluating all the proposals, ARPA selected BBN, a consulting firm in Cambridge, Massachusetts, and in December 1968, awarded it a contract to build the subnet and write the subnet software. BBN chose to use specially modified Honeywll DDP-316 minicomputers with 12k 16-bit words of core memory as the IMPs. The IMPs did not have disks, since moving parts were considered unreliable. The IMPs were interconnected by 56-kbps lines leased from telephone companies.

The software was split into two parts : **subnet** and **host**. The subnet software consisted of the IMP end of the host-IMP connection, the IMP-IMP protocol, and a source IMP to destination IMP protocol designed to improve reliability. The original ARPANET design is shown in Fig. 4.21.

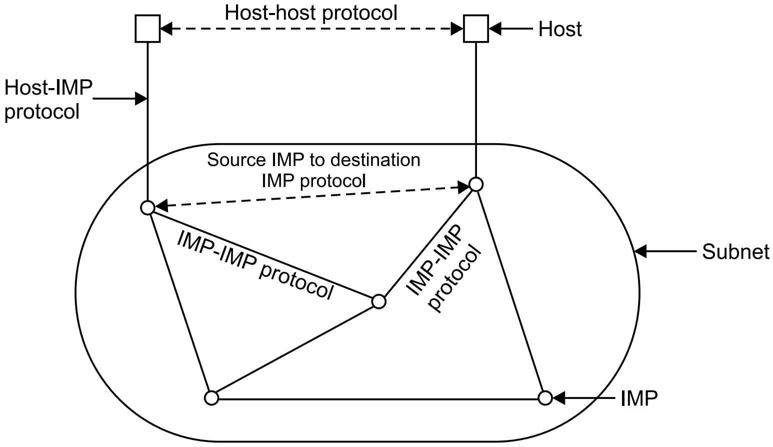


Fig. 4.21 The original ARPANET design.

Outside the subnet, software was also needed namely, the host end of the host-IMP connection, the host-host protocol, and the application software. It soon became clear that **BBN** felt that when it have accepted a message on a host-IMP wire and placed it on the host-IMP wire at the destinations, its job was done.

To deal with problem of host software, Larry Roberts of **ARPA** convened a meeting of network researchers, mostly graduate students, at Snowbird, Utah, in the summer of 1969. The graduate students expected some network expert to explain the design of the network and its software to them and then to assign each of them the job of writing part to it. They were astounded when there was no network expert and no grand design. They had to figure out what to do on their own.

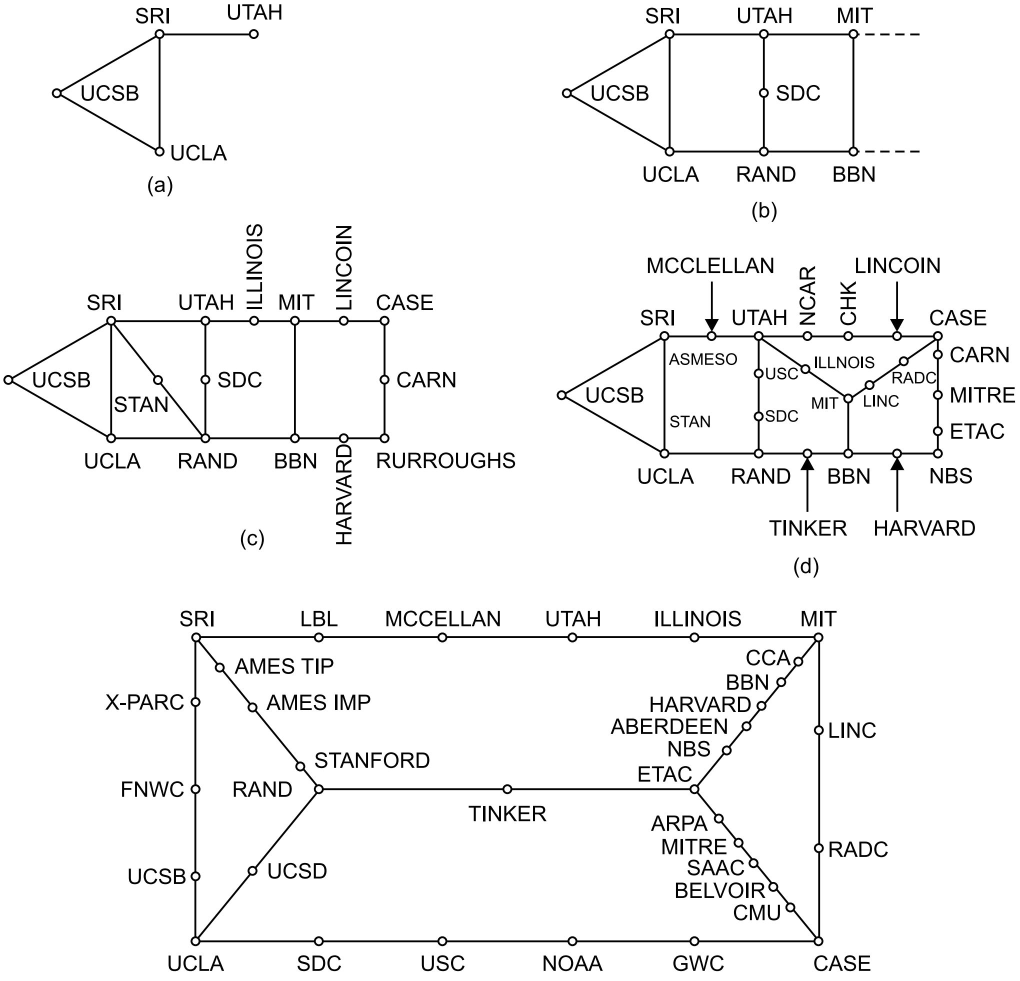


Fig. 4.22 Growth of the ARPANET (a) Dec. 1969 (b) July 1970 (c) March 1971 (d) April 1972 (e) Sept. 1972.

Nevertheless, somehow an experimental network went on the air in December 1969 with four node at UCLA, USCB, SRI and the University of Utah. These four were chosen because all had a large number of AROA Contract, and all had different and completely incompatible host computers (just to make it more fun). The network grew quickly as more IMPs were delivered and installed ; it soon spanned the United States. Figure 4.22 shows how rapidly the ARPANET grew in the first three years.

Later the IMP software was changed to allow terminals to connect directly to a special IMP, called a **TIP** (Terminal Interface Processor), without having to go through a host. Subsequent changes included having multiple hosts per IMP (to save money) hosts talking to multiple IMPs (to protect against IMP, failures), and hosts and IMPs separated by a large distance (to accommodate hosts far from the subnet).

In addition to helping the fledgling ARPANET grow, ARPA also funded research on satellite networks and mobile packet radio networks. In one famous demonstration, a truck driving around in California used the packet radio network to send messages to SRI, which were then forwarded over the ARPANET to the East Coast, where they were shipped to University College in London over the satellite network. This allowed a researcher in the truck to use a computer in London while driving around in California.

This experiment also demonstrated that the existing ARPANET protocols were not suitable for running over multiple networks. This observation led to more research on protocols, culminating with the invention of the TCP/IP was specifically designed to handle communication over internetworks, something becoming increasingly important as more and more networks were being hooked up to the ARPANET.

To encourage adoption of these new protocol ARPA awarded several contracts to **BBN** and the University of California at Berkeley to integrate them into Berkeley Unix. Researches at Berkeley developed a convenient program interface to the network (sockets) and wrote many applications utility, and management program to make networking easier.

The timing was perfect. Many Universities had just acquired a second or third **VAX** computer and a **LAN** to connect them, but they had no networking software. When 4.2 BSD came along with TCP/IP, sockets, and many network utilities the complete package was adopted immediately. Furthermore, with TCP/IP, it was easy for the LANs to connect to the ARPANET, and many did.

By 1983, the ARPANET was stable and successful, with over 200 IMPs and 100 of hosts. At this point, ARPA turned the management of the network over to the Defence Communication Agency **(DCA)**, to run it as an operational network. The first thing DCA did was to separate the military portion (about 160 IMPs, of which 110 in the United States and 50 abroad) into a separate subnet, **MILNET**, with stringent gateways between MILNET and the remaining research subnet.

The NSFNET

By the late 1970s, NSF (the U.S. National Science Foundation) saw the enormous impact the ARPANET was having on university research, allowing scientist across the country to share data and collaborate on research projects. However, to get on the ARPANET, a university had to have a research contact with the DoD, which many did not have. This lack of universal access prompted NSF to set up a virtual network, CSNET, centered around a single machine at BBN that supported dial-up lines and had connections to the ARPANET and other networks. Using CSNET, academic research could call up and leave email for other people to pick up later. It was simple, but it worked.

By 1984 NSF began designing a high-speed successor to the ARPANET that would be open to all university research groups. To have something concrete to start with, NSF decided to build a backbone network to connect its six supercomputer centres, in San Diego, Boulder, Champaign, Pittsburgh, Ithaca, and Princeton. Each supercomputer was given a little brother, consisting of an LSI-11 microcomputer called a **Fuzzball**. The fuzzballs were connected with 56 kbps leased lines and formed the subnet, the same hardware technology as the ARPANET used. The software technology was different however: the fuzzballs spoke TCP/IP right from the start, making it the first TCP/IP WAN.

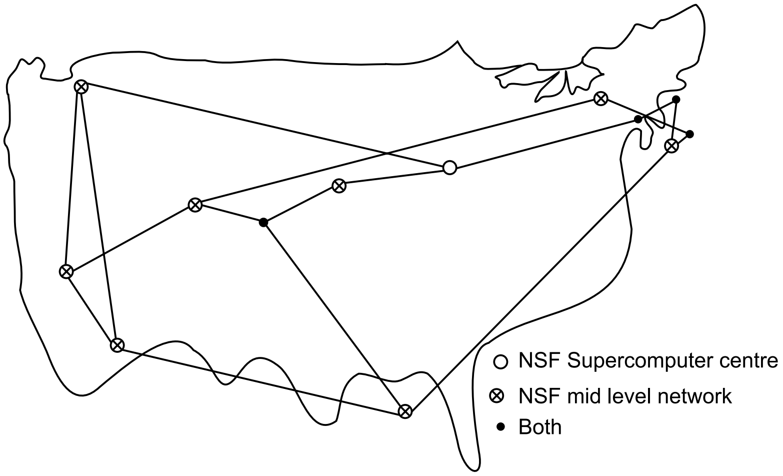


Fig. 4.23 The NSFNET backbone in 1988.

NSF also funded some (eventually about 20) regional networks that connected to the backbone to allow users at thousands of universities, research labs, libraries, and museums to access any of the supercomputers and to communicate with one another. The complete network, including the backbone and the regional networks was called **NSFNET**. It connected to the **ARPANET** through a link between an IMP and a Fuzzball in the Carnegie-Mellon machine room. The first **NSFNET** backbone is illustrated in figure 4.23.

NSFNET was an instantaneous success and was overloaded from the word go. NSF immediately began planning its successor and awarded a contract to the Michigan-based **MERIT** consortium to run it. Fiber optic channels at 448 kbps were leased from MCI to provide the version 2 backbone IBM RS 600s were used as routers. This, too, was soon overwhelmed, and by 1990, the second backbone was upgraded to 1.5 Mbps.

As growth continued, NSF realized that the government could not continue financing networking forever. Furthermore, commercial organizations wanted to join but were forbidden by NSF's charter from using network NSF paid for. Consequently, NSF encouraged **MERIT**

MCI, and IBM to form a nonprofit corporation, **ANS** (Advanced Networks and Services) as a step along the road to commercialization. In 1990, ANS took over **NSFNET** and upgraded the 1.5 Mbps links to 45 Mbps to form **ANSNET**.

The INTERNET

The number of networks, machines and users connected to the **ARPANET** grew rapidly after TCP/IP became the only official protocol on Jan. 1, 1983. When NSFNET and the ARPANET were interconnected, the growth became exponential. Many regional networks joined up, and connections were made to networks in Canada, Europe and the Pacific.

Sometime in the mid-1980s, people began viewing the collection of networks as an internet, and later as the Internet, although there was no official dedication with some politician breaking a bottle of champagne over a fuzzball.

Growth continued exponentially, and by 1990 the Internet had grown to 3000 networks and 200,000 computers. In 1992, the one millionth host was attached. By 1995, there were multiple backbones, hundreds of mid level (i.e., regional) networks, tens of thousands of LANs, millions of hosts, and tens of millions of users. The size doubles approximately every year (Paxson, 1994).

Much of the growth comes from connecting existing networks to the Internet. In the past these have included DPAN, NASA's space physics network, HEPNET, a high energy physics network, BITNET, IBM's mainframe network, EARN, a European academic network now widely used in Eastern Europe, and many others. Numerous transtlantic links are in use; running from 64 kbps to 2 Mbps.

The glue that holds the Internet together is the TCP/IP reference model and TCP/IP protocol stack. TCP/IP makes universal service possible and can be compared to the telephone system or the adoption of standard gauge by the railroads in the 19th century.

What does it actually mean to the on the Internet ? Our definition is that a machine is on the Internet if it runs the TCP/IP protocol stack, has an IP address, and has the ability to send IP packets to all the other machines on the Internet. The mere ability to send and receive electronic mail is not enough, since email is gatewayed to many networks outside the Internet. However, the issue is clouded somewhat by the fact that many personal computers have the ability to call up an Internet service provider using a modem, be assigned a temporary IP address and send IP packets to other Internet hosts. It make sense to regard such machines as being on the Internet for as long as they are connected to the service provider's router.

1. Email

The ability to compose, send, and receive electronic mail has been around since the early days of the ARPANET and is enormously popular. Many people get dozens of messages a day and consider it their primary way of intracting with the outside world, far outdistancing the telephone and snail mail. Programs are available on virtually every king of computer these days.

2. News

News groups are specialized forms in which users with a common interest can exchange messages. Thousands of new groups exist, on technical and nontechnical topics, including computers, science, recreation, and politics. Each news group has its own etiquette, style, and customs, and woe be to anyone violating them.

3. Remote Login

Using the Telnet, Rlogin, or other programs, users any where on the Internet can log into any other machine on which they have an account.

4. File Transfer

Using the FTP program, it is possible to copy files from one machine on the Internet to another. Vast numbers of articles databases and other information are available this way.

Up until the early 1990s, the Internet was largely populated by academic, government and industrial researchers. One new application, the **WWW (World Wide Web)** changed all that and brought millions of new, nonacademic users to the net. This application, invented by CERN physicist **Tim Berners-Lee,** did not change any of the underlying facilities but made them easier to use. Together with the Mosaic viewer, written at the National center for supercomputer application containing text, pictures, sound, and even video, with embedded links to other pages. By clicking on a link, the user is suddenly transported to the page entries pointing to other pages for product information, price lists, sales, technical support, communication with employees, stockholder information, and much more.

Numerous other kinds of pages have come into existence in a very short time, including maps, stock market table, library card catalogs, recorded radio programs, and even a page pointing to the complete text of many books whose copyrights have expired. Many people also have personal pages (home pages).

GIGABIT TESTBEDS

The Internet backbone operate at megabit speeds, so for people who want to push the technological envelope, the next step is gigabit networking. With each increase in network bandwidth new applications become possible, and gigabit networks are no exception. In this section we will first say a few words about gigabit applications, mention two of them, and then list some example gigabit testbeds that have been built.

Gigabit networks provide bandwidth than megabit networks, but not always much better delay. For example, sending a 1 Kb packet from New York to San Franscisco of 1 Mbps takes 1 msec to pump the bits out and 20 msec for the transcontinental delay. For a total of 21 msec. A 1 Gbp network can reduce this to 20.001 msec. While the bits go out faster, the transcontinental delay remains the same, since the speed of light in optical fiber (or copper wire) is about 200,000 km independent of the data rate. Thus for wide area applications in which low delay is critical, going to higher speeds may not help much. Fortunately, for some applications, bandwidth is what counts and these are the applications for which gigabit networks will make a big difference.

On application is telemedicine. Many people think that a way to reduce medical costs is to reintroduce family doctors and family clinics on a large scale, so everyone has convenient access to first line medical care. When a serious medical imaging, such as x-rays, CAT scans, and MRI scans. The test results and images can then be sent electronically to a specialist who then make the diagnosis.

Doctors are generally unwilling to make diagnoses from computer images unless the quality of the transmitted image is as good as the origin image. This requirement means images will probably need 4K × 4K pixels, with 8 bits per pixel (black and white images) or 24 bits per pixel (color images). Since many tests require up to 100 images (*e.g.,* different cross sections of the organ in question), a single series for one patient can generate 40 gigabits. Moving images (*e.g.,* a beating heart) generate even more data. Compression can help some but doctors are leary of it because the most efficient algorithms reduce image quality. Furthermore, all the images must be stored for years but may need to be retrived at a moment's notice in the event of a medical emergency. Hospitals do not want to become computer centers, so off-site storage combined with high-bandwidth electronic retrival is essential.

Another gigabit application is the **virtual meeting.** Each meeting room contains a spherical camera and one or more people. The bit streams from each of the cameras are combined electronically to give the illusion that everyone is in the same room. Each person sees this image using virtual reality goggles. In this way meeting can happen without travel, but again, the data rates required are stupendous.

QUESTIONNARIES

1. What is computer network? Give its advantages and applications.
2. Discuss the factors that affect the performance and reliability of the network.
3. Explain the communication procedure between the layers. What different units exist in this communication?
4. Explain various issues in designing of layers in layered reference models and explain term protocol.
5. Explain OSI reference model.
6. Draw TCP/IP model and explain function of each layer.
7. Compare OSI and TCP/IP reference model.
8. Explain different network topologies.
9. Explain ATM reference model.

Chapter 5

NETWORK CONCEPTS AND COMPONENTS

INTRODUCTION

In this chapter we will study various networking concepts as wireless networks, Networks Interfaces, different services offered by the Network and various protocols.

Also we will be discussing about various networking components as cabling and connector std., NIC, Hubs, Bridges, etc.

5.1 NETWORK CONCEPTS

Networking concepts consists of layered approach, different Interfaces and variety of services provided by them. Protocol functions and brief about X.25 protocol, Intranet and Extranets.

5.1.1 Wireless Networks

Mobile computers, such as notebook computers and personal digital assistants (PDAs), are the fastest growing segment of the computer industry. Many of the owners of these computers have desktop machines on LANs and WANs back at the office and wants to be connected to their home base even when away from home or route. Since having a wired connection is impossible in cars and air planes. There is a lot of interest in wireless networks.

Actually, digital wireless communication is not a new idea. As early as 1901, the Italian physicist Guglielmo Marconi demonstrated a ship to store wireless telegraph using morse code. Modern digital wireless systems have better performance, but the basic idea is the same.

Wireless networks have many uses. A common one is the portable office. People on the road often want to use their portable electronic equipment to send and receive telephone calls, faxes and electronic mail, read remote files, login on remote machines, and so on, and do this from anywhere on land, sea, or air.

Wireless networks are of great value to fleets of trucks, taxies, buses and repair persons for keeping in contact with home. Another use is for rescue worker at disaster sites (fires, floods, earthquakes, etc., where the telephone system has been destroyed. Computer there can send messages, keep records and so on.

Finally, wireless networks are important to the military. If you have to be able to fight a war anywhere on earth on short notice, counting on using the local networking infrastructure is probably not a good idea. It is better to bring your own.

Although, wireless networking and mobile computing are often related, they are not identical, which is shown in Table 5.1 portable computers are sometimes wired.

Table 5.1 Combination of wireless networks and mobile computing.

Wireless	*Mobile*	*Applications*
No	No	stationary workstations in offices.
No	Yes	using a portable in hotel.
Yes	No	LANs in older, unwired bldg.
Yes	Yes	Portable office.

On the other hand, some wireless computers are not portable. But of course, there are also the true mobile, wireless applications, ranging from the portable office to people walking around a store with a PDA doing inventory. At many busy airports, car rental return clerks work out in the parking lot with wireless portable computers. They type in the license plate number of returning cars, and their portable, which has a built in printer, calls the main computer gets the rental information and prints out the bill on the spot.

Wireless networks come in many forms. Some universities are already installing antennas all over campus to allow students to sit under the trees and consult the library's card catalog. Here the computers communicate directly with the wireless LAN in digital form. Another possibility is using a cellular telephone with a transitional analog modem. Direct digital cellular services, called CDPD (Cellular Digital Packet Data) is becoming available in many cities.

Finally, it is possible to have different combinations of wired and wireless networking for example, the flying LAN, a traditional LAN.

5.1.2 Layered Approach

Some of the key design issues that occur in computer networking are present in several layers.

(1) Every layer needs a mechanism for identifying senders and receivers. Since network normally has many computers, some of which have multiple processors means is needed for a process on one machine to specify with whom it wants to talk. As a consequence having multiple destinations, some form of addressing is needed in order to specify a Specific destination.

(2) Another set of design decision concerns the rules for data transfer, such as.

1. Simplex 2. Half duplex 3. Full duplex.

(3) Error control is an important issue because physical communication circuits are not perfect. Many error-detecting and error-correcting codes are known, but both ends of the connection must agree on which one is being used. Also the receiver must have some way of telling the sender which messages have been correctly received and which have not.

(4) Not all communication channels preserve the order of messages sent on them. To deal with a possible loss of sequencing the protocol must make explicit provision for the receiver to allow the pieces to be put back together properly.

(5) An issue that occurs at every level is how to keep a fast sender from swamping a slow receiver with data.

(6) Another problem that must be solved at several levels is the inability off all processes to accept arbitrarily long messages. This property leads to mechanisms for disassembling, transmitting and then reassembling messages.

When there are multiple paths between source and destination, a route must be chosen sometimes this decision must be split over two or more layers.

5.1.3 Interfaces

The function of each layer is to provide services to the layer above it.

The active elements in each layer are often **called entities.** An entity can be a software entity (such as a process), or a hardware entity (such as an intelligent I/O chip). Entities in the same layer on different machines are called **peer entities.** The entities in the layer *n* implement a service used by layer *n* + 1. In this case layer *n* is called the **service provider** and layer *n* + 1 is called the **service user.** Layer *n* may use the services of layer *n* – 1 in order to provide its services. It may offer several classes of service for example fast, expensive communication and slow, cheap communication.

Services are available at SAPs (Service Access Points). The layer *n* SAPs are the places where layer *n* + 1 can access the services offered. Each SAP has an address that uniquely identifies it. To make this point clear, the SAPs in the telephone system are the sockets into which modular telephone can be plugged, and the SAP addresses are the telephone numbers of these sockets. To call someone, you must know the caller's SAP address. Similarly, in the postal system, the SAP addresses are street addresses and post office box numbers. To send a letter, you must know the addresses.

SAP = Service Access Point

IDU = Interface Data Unit

SDU = Service Data Unit

PDU = Protocol Data Unit

ICI = Interface Control Information

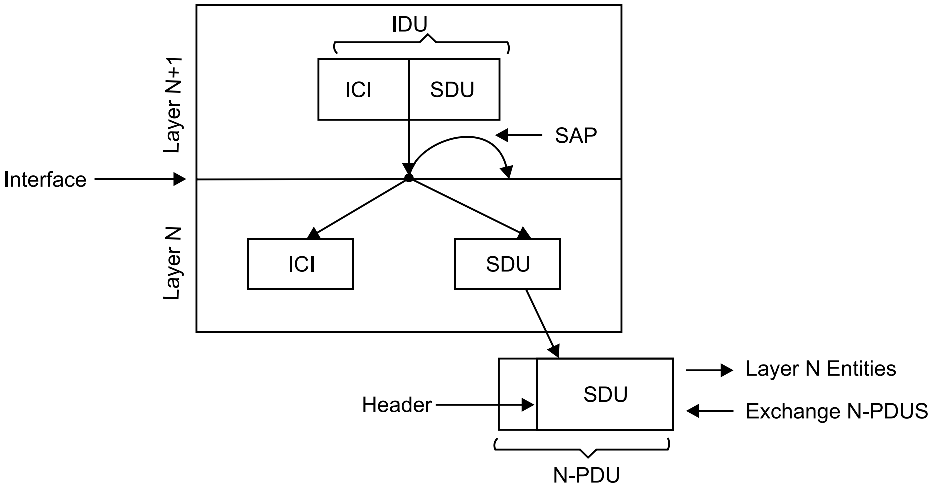


Fig. 5.1 Relation between layers at an interface

In order for two layers to exchange information, there has to be an agreed upon set of rules about the interface. At a typical interface, the layer *n* + 1 entity passes an IDU (Interface Data Unit) to the layer *n* entity through the SAP as shown in the Fig. 5.1. The IDU consists of an SDU (Service Data Units) and some control information. The SDU is the information passed across the networks to the peer entity and then up to layer *n* + 1, the control information is needed to help the lower layer do its job but is not part of the data itself.

In order to transfer the SDU, the layer *n* entity may have to fragment it into several pieces, each of which is given a header and sent as a separate PDU (Protocol Data Unit) such as a packet. The PDU headers are used by the peer entities to carry out their peer protocol. They identify which PDUs contain data and which contain control information, provide sequence number and counts, and so on.

5.1.4 Services

Layers offer two types of services.

1. Connection Oriented Service

This is modeled after the telephone system. To talk to some one, you pick up the phone, dial the number, talk and then hang up. Similarly, to use a connection oriented network services, the services user first establishes a connection, uses the connection and then releases the connection. The essential aspect of connection is that it acts like a tube. The sender pushes objects (bits) in at one end and the receiver takes them out in the same order at the other end.

Connection oriented services are of following:

1. Reliable message stream service.

   e.g. sequence of pages.

2. Reliable byte stream.

   e.g. remote login.

3. Unreliable connection.

   e.g. Digitized voice.

2. Connectionless Services

This is modeled after the postal system. Each message carries the full destination address and each one is routed through the system independent of all the (users) others.

Normally, when two messages are sent to the same destination, the first one sent will be the first one to arrive. However, it is possible that the first one sent, can be delayed so that the second one arrives first. With connection oriented service this is impossible.

Connectionless services are of the following:

1. Unreliable datagram.

   e.g. Electronic junk mail.

2. Acknowledged datagram.

   e.g. Registered mail.

3. Request reply service.

   e.g. Database query.

5.1.5 Protocols

A protocol is used for communication between entities in different systems. The terms 'entity' and 'system' are used in a very general sense. Examples of entities are user application programs, file transfer package, database management systems, electronic mail facilities, and terminals.

Examples of systems are computers, terminals, and remote sensors. Note that in some cases the entity and the system in which it resides are coextensive (*e.g.* terminal).

In general, an entity is anything capable of sending or receiving information, and a system is physically distinct object that contains one or more entities. For two entities to communicate successfully, they must 'speak the same language'. What is communicated? How it is communicated and when it is communicated? Must conform to some mutually acceptable conventions between the entities involved. The conventions are referred to as a protocol.

Protocol may be defined as a set of rules governing the exchange of data between two entities. It also may be conceived as a set of agreement between two communicating processes.

The key elements of protocol are:

- **Syntax:** Includes such things as data format and signal levels.
- **Semantics:** Includes control information for co-ordination and error handling.
- **Timing:** Includes speed matching and sequencing create new paragraph. Having introduced the concept of protocol, we can now introduce the concept of a protocol

architecture. It is clear that there must be a high degree of co-operation between the two computers. Instead of implementing the logic for this as a single module, the task is broken up into subtasks, each of which is implemented separately. As an example figure. 5.2 suggests the way in which a file transfer facility could be implemented.

Let us try to summarize the motivation for the three module in Fig. 5.2. The file transfer module contains all of the logic that is unique to the file transfer application, such as transmitting passwords, file commands, and file records. There is a need to transmit these files and commands reliably. However, the same sort of reliability requirements are relevant to a variety of applications. The communications service module is concerned with assuring that the two computer systems are active and ready for data transfer, and for keeping track of the data that are being exchanged to assure delivery. However, these tasks are independent of the type of network that is being used. Therefore, the logic for actually dealing with the network is separated out into a separate network access module. That way, if the network to be used is changed, only the network access module is affected.

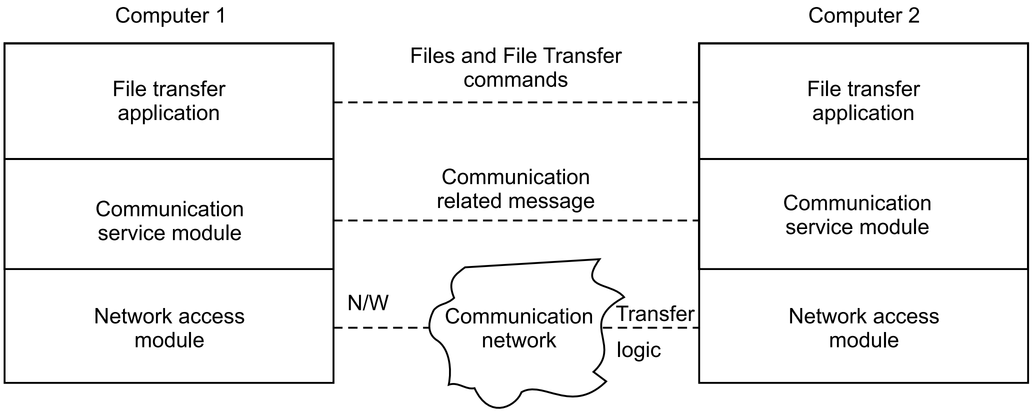

Computer 1
Computer 2
File transfer application
Files and File Transfer commands
File transfer application
Communication service module
Communication related message
Communication service module
Network access module
N/W
Communication network
Transfer
logic
Network access module

Fig. 5.2 A simplified architecture for file transfer

Thus, instead of a single module for performing communications, there is a structured set of modules that implements the communications function. That structure is referred to as a protocol architecture.

Functions of protocols

Some of the numerous functions served by network protocol are as follows:

- Orderly exchange of data messages.
- Management of priorities at both the network entry and transmission levels within the network.
- Process synchronisation.
- Session establishment between network users.

- Session termination between network users.
- Means for protocol validation.
- Routing establishment and assignment of message routes and routing information.
- Flow control and congestion prevention.
- Sequencing—sequenced transmission and delivery of messages.
- Addressing of network components and users.
- Efficient network resources utilisation.
- Resource management, monitoring and protection.
- Layered transparency between networks users and nodes.
- Reliable message transmission, including error, control and recovery.
- Testing of network resources, such as links and routes.
- Security and privacy.
- Optional packet switching through message segmenting and pipelining.

5.1.6 Brief Study of X.25 Protocol

In the early 1970's there were many data communication networks, which were owned by private companies, organisations and government agencies. Since those public networks were quite different internally and the inter connection of networks was growing very fast, there was a need for a common network interface protocol.

In 1976's the international consultative committee recommend X.25 as the desired protocol for telegraphy and telephony (CCITT) called the International Telecommunication Union (ITU) since 1993.

X.25 is a packet switched data network protocol which defines an international recommendation for the exchange of data as well as control information between a user device (host), called Data Terminal Equipment (DTE) and a network node, called Data Circuit Terminating Equipment (DCE).

The X.25 protocol adopted as a standard in the 1970's by CCITT is a commonly used network protocol. X.25 protocol allows computer on different public networks to communicate through and intermediary computer at the network layer. It uses PSPDN (Packet Switched Public Data Network) as the carrier and needs to have PAD (Packet Assembler Disassembler) at both ends. X.25 packet switched networks allow remote devices to communicate with each other, across high speed digital link without individual leased lines. X.25 is connection oriented and supports both switched virtual circuits (SVC) and permanent virtual circuits (PVC). In X.25 packet switched public data network, the equipment that are used at user end are DTE (Data Terminal Equipment) and DCE (Data Communication Equipment). DTE/DCE equipment include, host computer with interface, PAD (Packet Assembler, Disassembler), and gateway between LAN and PDN (Public Data Network). Packet switched networks use virtual circuits.

There are two types of virtual circuits.

1. Switched Virtual Circuit (SVC).
2. Permanent Virtual Circuit (PVC).

1. Switched Virtual Circuit (SVC)

SVC is created when one computer send a packet to the network requesting to make a call to remote computer. Once the connection is established, the packets can be sent over that connection. After the data transmission, the connection is released. The data is transmitted on the basis of unique address assigned to each DTE (Data Terminal Equipment).

2. Permanent Virtual Circuit (PVC)

PVC is similar to leased lines that the permanent and virtual except that the customer pays only the time the line is used. No initial call setup is required that data can be sent at any time. A permanent logical connection is established the packet switched network administration. This is shown in Fig. 5.3.

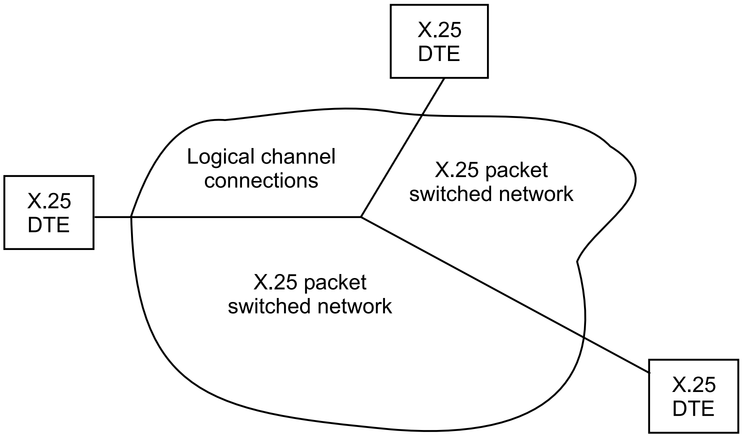


Fig. 5.3 X.25 packet switched network

X.25 packet switched networks allow remote devices to communicate with each other across high speed digital links without the expense of individual leased lines. Packet switching is a technique whereby the network routes individual packets of HDLC data between different destinations based on addressing within each packet. X.25 utilies a connection oriented service insures that packets are transmitted in order.

X.25 comes with three levels based on the first three layers of the Open System Inter connection (OSI) seven layer architecture as defined by the International Standards Organisation (ISO). This is illustrated in Fig. 5.4.

The three levels are:

1. ***The physical level:*** The physical level describes the interface with the physical environment. It is similar to the physical layer in the OSI models.

2. ***The link level:*** The link level is responsible for the reliable communication between the DTE and the DCE. It is similar to the data link layer in the OSI model.
3. ***The packet level:*** The packet level describes the data transfer protocol in the packet switched network. It is similar to the network layer in the OSI model.

X.25 was originally approved in 1976 and subsequently revised in 1977, 1980, 1984, 1988 and 1992. It is currently (1996) one of the most widely used interfaces for data communication networks. X.25 is an analog packet switching networks. It can be considered slow packets switching. The transfer speeds are typically 56 kbps to 2.08 Mbps. There is a worldwide set of public X.25 networks and it is possible for an organisation to have its own private X.25 network.

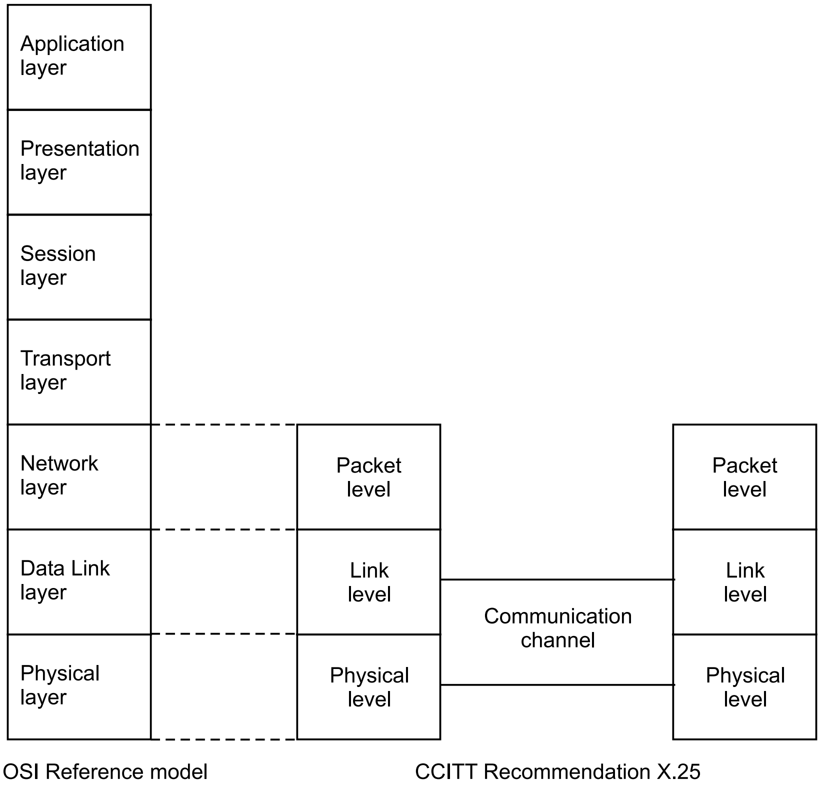


Fig. 5.4. CCITT recommendation X.25 model

X.25 is over 20 years old and an established technology. There are many multivendor solutions and dissimilar technologies in an organisation are allowed to access the X.25 network. In Canada, the main X.25 Network Datapac that is a public offering of X.25. You pay either a flat rate or by the packet. X.25 is used to connect LAN together. Due to its slow transfer speed, it is used for the following:

- Host terminal emulation; low data.
- Client/server application such as E-mail; small files, bandwidth.

- File server; large amount of data and real time traffic.
- Database; usually large databases but queries are small inbound and medium size outband.

X.25 is a high speed overhead compared to other networks. This reduces the transfer speed and bandwidth utilisation meaning that it is not as efficient.

Let us now examine each level in more detail.

The Physical Level

The physical level (level 1) deals with the electrical, mechanical, procedural and functional interface between the DTE and the DCE.

X.21, X.21-bis and the V.24 recommendation for modems and interchange circuits specify the physical level.

X.21

X.21 interface is a CCITT recommendation for operation of digital circuits. The X.21 interface operates over eight interchange circuit (*i.e.* signal ground, DTE common return, transmit, receive, control indication, signal element aiming and byte timing) their functions defined by recommendation X.24 and their electrical characteristics in recommendation X.27.

X.21-bis

X.21-bis is a CCITT recommendation that defines the analog interface to allow access to the digital circuit switched network using an analogue circuit X.21-bis provides procedure for sending and receiving addressing information, which enable a DTE to establish switched circuits with other DTEs, which have access to the digital network.

V.24

V.24 is also a CCITT recommendation. It provides procedure that enable the DTE to operate over a leased analogue circuit connecting it to a packet switching node or connector.

The Link Level

The link level (also called level 2, or frame level) ensures reliable transfer of data between the DTE and the DCE, by transmitting the data as a sequence of frames (A frame is an individual data unit which contains address, control, information field etc.).

The function performed by the link level includes:

- Transfer of data in an efficient and timely fashion.
- Synchronisation of the link to ensure that the receiver is in step with the transmitter.
- Detection of transmission errors and recovery from such errors.
- Identification and reporting of procedural errors to higher levels, for recovery.

The data link level uses link control procedures, which are compatible with the High Level Data Link (HDLC) standardised by ISO, and with the Advanced Data Communications Control Procedures (ADCCP) standardised by the American National Standards Institute (ANSI).

The several protocol that can be used in the link level are as follows:

Link Access Protocol (LAP)

LAP is an earlier version of LAPB and is seldom used today.

Link Access Protocol Balanced (LAPB)

LAPB is derived from HDLC and is the most commonly used. It enables to form a logical link connection besides all the other characteristics of HDLC.

Link Access Procedure, D channel (LAPD)

LAPD is received from LAPB and it is used for integrated services. Digital Networks (ISDN) *i.e.*, it enables data transmission between DTEs through D channel, especially between a DTE and an ISDN node.

Logical Link Control (LLC)

LLC is an IEEE 802 local area network (LAN) protocol that enables X.25 packet to be transmitted through a LAN channel.

The Packet Level

The packet level (also called level 3 or network level) create network data unit called packets that contain control information and user data.

Packet level provides procedures for handling the following services:

Virtual Circuit (VC)

VC is a temporary association between two DTEs, it is initiated by a DTE signaling a CALLREQUEST to the network. This service ensures orderly (sequenced) delivery of packet in either direction between two DTEs. Virtual circuit is established whenever two DTEs want to communicate. This service is the most frequently used by the X.25.

Permanent Virtual Circuit

Permanent virtual circuit is a permanent association existing two DTEs, which do not require call setup (connect) or call clearing (disconnect) action by the DTEs.

Datagram (DG)

DG is a self contained user data unit, containing sufficient information to be routed to the destination DTE with a need of a call to be established.

The data units are transmitted one at a time with no guarantee of final delivery, and no guarantee of orderly delivery. Each datagram must contain complete address. Procedures and control information to enable it to be delivered to the proper destination DTE.

Fast Select

Fast select is a service that enables the control packet that setup the VC to carry data as well.

Other Services

The packet level also provides the call setup and call clearing procedures required for the VC service.

The packet level deals with flow control to ensure that a user (DTE) does not overwhelm the other user with packets, and to maintain timely and efficient delivery of packets. The packet level to abort restarts VCs if necessary.

Following features that make X.25 unique include:

- X.25 defines the procedures for the exchange of data between user devices (DTEs) and a packet network node (DCE).
- X.25 provides flow and error control functions.
- The procedures includes functions such as identifying packets of specific user terminals and computers, acknowledgment packets, rejecting packets and providing for error recovery and flow control.
- X.25 also provides some useful facilities such as the changing of packets to DTE stations other than transmitting DTEs.
- X.25 standard contains no routing algorithm.
- X.25 operator on the premise of virtual circuit services.
- X.25 uses logical channel number to identify the DTE connection to the network.
- X.25 allows the user device (DTE) or packet exchange to limit the rate at which it accepts packets.

X.25 State diagrams

The X.25 standard contains several state diagrams to describe event sequence, such as a call setup and call clearing. Fig. 5.5 shows the subphases of call setup.

1. DTE — CALL REQUEST
2. DCE — CALL CONNECTED
3. DCE — INCOMING CALL
4. DTE — CALL ACCEPTED

5. DCE — INCOMING CALL
6. DTE — CALL REQUEST
7. DCE — CALL CONNECTED

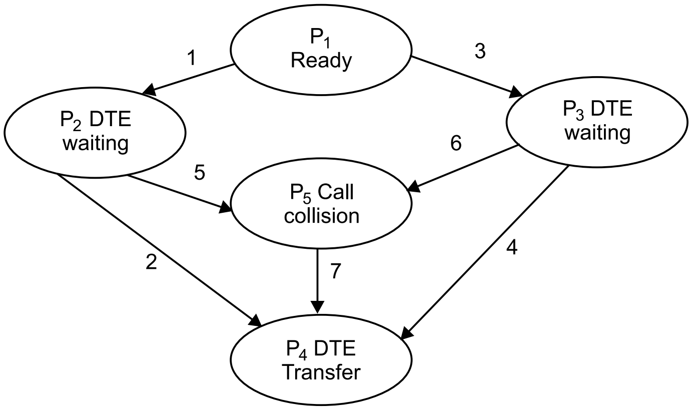


Fig. 5.5 State diagram of call setup

Initially, the interface is in state P_1. A CALL REQUEST or INCOMING CALL changes the state to P_2 or P_3, respectively. From these states the data transmit state, P_4 can be reached, either directly or via P_5.

Similar diagrams are provided for call clearing, resetting and restarting.

Advantages of X.25

1. Virtual circuit switching and dynamic virtual routing to transport self contained, self addressed message packet.
2. Ability to use any available network channel or link.
3. Uses redundant error checking at every nodes.
4. Store and forward nature of packet switching DTEs do not need to have the same speed as other DTEs with which it communicates. Hence, speed matching prior to transmission is not necessary.
5. Provides a virtual high quality networks at low costs.

Disadvantages of X.25

1. X.25 does have some drawbacks. There is an inherent delay caused by the store and forward mechanism. On most single network the turn around delay is about 0.6 seconds. This has no effect on large block transfer, but in flip-flop types of transmissions the delay can be very noticeable.

2. X.25 is a data pump, there has to be some higher level that is making sense of the bits.

5.1.7 Intranet and Extranet

Intranet

An Intranet is an organisations own internal network that uses the same services, protocols and technologies which are available on the Internet.

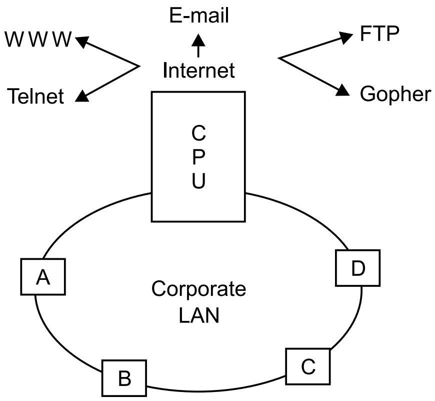


Fig. 5.6 Intranet

Users use the same software to utilise services an the Internet as well as Intranet.

The Intranet can provides all the services that the Internet provides, but only within the organisation.

- *E-mail:* E-mail used for sending/receiving mails (SMTP, POP$_3$).
- *PTP:* File transfer between internetwork nodes.
- *Gopher:* Gopher system used for organising and displaying files on internet.
- *Telnet:* Telnet used for remote terminal emulation.
- *WWW*: World Wide Web, which allows users to create, edit and browse the hypertext documents.

Extranet

When an organisation's, Intranet is made available to selected users outside its local area network, it is known as **extranet**.

It provides better communication and collaboration with customers.

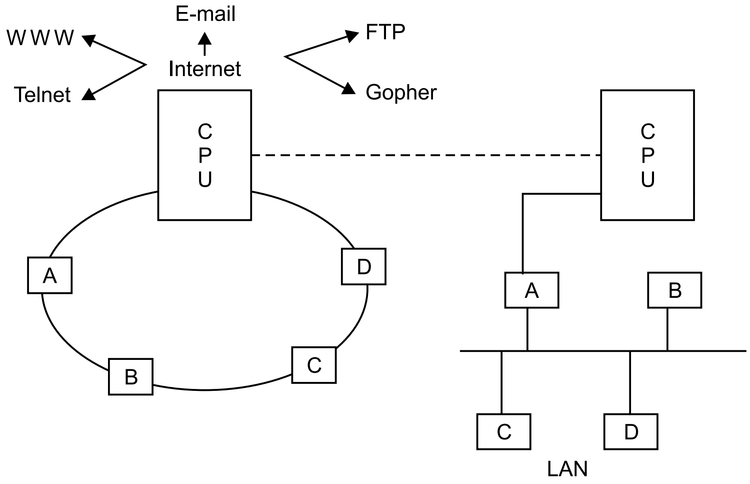


Fig. 5.7 Extranet

5.2 NETWORK COMPONENTS

In this we will examine many of the different hardware components involved in networking like NIC, hubs, Routers different cables, etc.

5.2.1 Cabling and Connector Standards

The IEEE 802.3 committee has been the most active in defining alternative physical configuration. This is both good or bad.

On the good side, the standard has been responsive to evolving technology. Thus the user that has a complex setof requirements may find the flexibility and variety.

The defined alternatives are:

1. 10 BASE 5
2. 10 BASE 2
3. 10 BASE-T
4. 10 BROAD 36
5. 10 BASE-F

Table 5.2 10 Mbps physical layer medium alternative.

Parameter	*10 Base 5*	*10 Base 2*	*10 Base T*	*10 Broad 36*	*10 Base F*
Transmission Medium	Coaxial cable (50 ohm)	Coaxial cable (50 ohm)	Unshielded Twisted pair	Coaxial cable (75 ohm)	850-nm Optical fiber
Signalling Technique	Baseband Manchester	Baseband Manchester	Baseband Manchester	Broadband (DPSK)	Manchester on-off
Topology	BUS	BUS	STAR	BUS/TREE	STAR
Max. segment length (m)	500	185	100	1800	500
Cable diameter (mm)	10	5	0.4–0.6	0.4–1	62.5/125–400
Nodes/segment	100	30	–	–	33

10 BASE 5

10 BASE 5 is the original 802.3 medium specification and is based on directly on Ethernet. 10 BASE 5 specifies the use of 50-ohm co-axial cable and uses Manchester digital signalling.

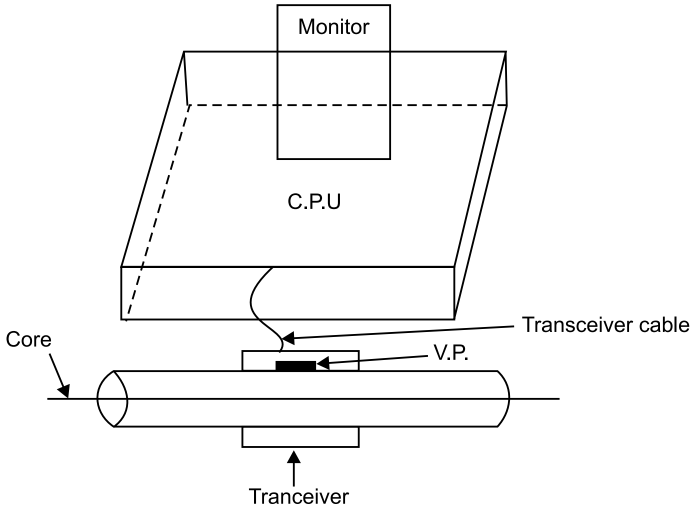


Fig. 5.8 10 BASE 5

RG-11 Co-axial cable, used for 10 BASE 5 operation (also called thickwire). The maximum length of a cable segment is set at 500 m. The length of the network can be extended by the use of repeaters, which are transparent to the MAC level, as they do not buffering, they do not isolate one segment from another.

Example: If two stations on different segments attempt to transmit at the same time, their transmission will collide. To avoid looping, only one path of segments and repeaters allowed between any two stations.

The standard allows a maximum of four repeaters in the path between any two stations, there by extending the effective length of medium to 2.5 km.

10 BASE 2

To provide a lower cost systems than 10 BASE 5 for personal computer, LANs, 10 BASE 2 was needed.

RG-58 coaxial cable, used for 10 base 2 operation (also called thinwire).

As with 10 BASE 5, this specification uses 50-ohm coaxial cable and Manchester signalling. The key difference is that 10 BASE 2 uses a thinner cable, which supports fewer taps over a shorter distance than the 10 BASE 5 cable.

Because they have the same data rate, it is possible to combine 10 BASE 5 and 10 BASE 2 segment in the same network by using a repeater that conforms to 10 BASE 5 on one side and 10 BASE 2 on the otherside. The only restriction is that a 10 BASE 2 segment should not be used to bridge two 10 BASE 5 segments, because a "backbone" segment should be as resistant to noise as the segments it connects.

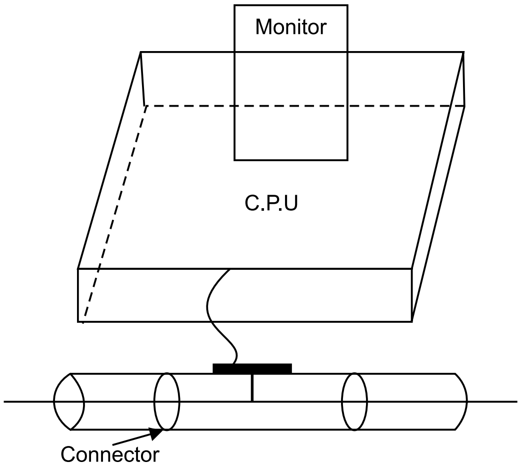


Fig. 5.9 10 BASE 2

10 BASE-T

By sacrificing some distance, it is possible to develop a 10 Mbps LAN using the unshielded twisted pair medium. Such wire is often found prewired in office building as excess telephone cable and can be used for LANs.

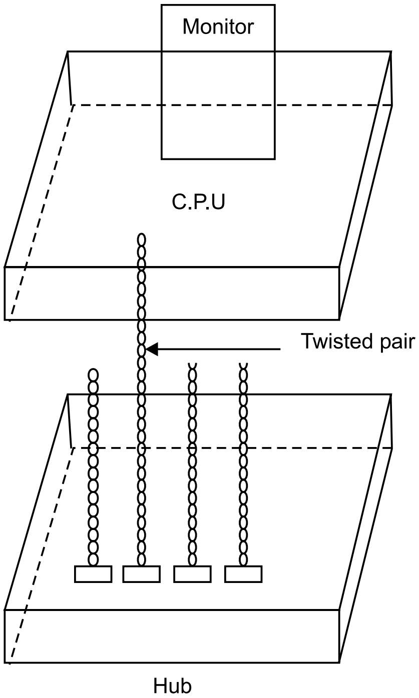


Fig. 5.10. 10 BASE–T

Such an approach is specified in the 10 BASE-T specification. The BASE-T specification defines a star shaped topology.

A simple system consists of a number of stations connected to a central point, refered to as a multipoint repeater, via two twisted pairs. The central point accepts input on any one line and repeats it on all of the other lines.

Stations attach to the multiport repeater via a point to point link. Ordinarily the link consists of two unshielded twisted pairs. Because of the high data rate and the poor transmission quality of unshielded twisted pair, the length of a link is limited to 100 m.

10 BROAD 36

The 10 BROAD 36 specification is the only 802.3 specification for broadband. The medium employed is the standard 75 ohm. CATV coaxial cable, either a dual cable or split cable configuration is allowed.

The maximum length of an individual segment, emanating from the headend is 1800 m. This results in a maximum end to end span of 3600 m.

The signalling on the cable is differential phase shift keying (DPSK). In ordinary PSK, binary zero is represented by a carrier with a particular phase and a binary one is represented by a carrier with the opposite phase (180° difference) DPSK makes use of differential encoding, in which a change of phase occurs when a zero occurs.

The advantage of differential encoding is that it is easier for the receiver to detect a change in phase than to determine the phase itself.

The characteristics of the modulation process are specified so that the resulting 10 Mbps signal fits into a 14 Mbps bandwidth.

10 BASE-F

The 10 BASE-F enables users to take advantage of the distance and transmission characteristics available with the use of optical fiber.

They have following three standards:

1. 10 BASE-FP (Passive)
2. 10 BASE-FL (Link)
3. 10 BASE-FB (Backbone)

All these three make use of a pair of optical fibers for each transmission link, one for transmission in each direction. In all cases the signalling scheme involves the use of Manchester encoding. Each Manchester signal element is then converted to an optical signal element with the presence of light corresponding to high and the absence of light corresponding to low. Thus a 10 Mbps Manchester bit stream actually requires 20 Mbps on the fiber.

1. 10 BASE-FP

A passive star topology for interconnecting stations and repeaters with up to 1 km/segment.

It defines a passive star system that can support up to 33 stations attached to a central passive star.

It also defines point to point connections that can be used to extend the length of a network.

2. 10 BASE-FL

It defines a point-to-point link that can be used to connect stations or repeaters at up to 2 km.

A conventional asynchronous signalling is used with 10 BASE-FL, No such retimings takes place, so that any timing distortions are propagated through a series of repeaters.

3. 10 BASE-FB

It defines a point to point link that can be used to connect repeaters at up to 2km.

It can also be used to cascade up to 15 repeaters in sequence to achieve greater length.

Cable Topologies

Cable topologies are of four types.

1. *Linear (BUS) topology:* A single cable is snaked from 1.

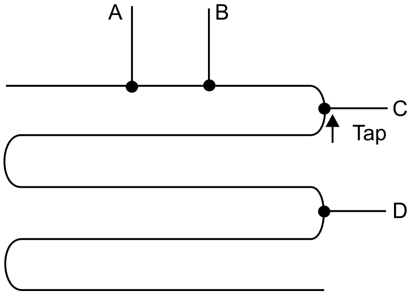


Fig. 5.11 Bus topology

2. *Spine topology:* A vertical spine runs from the basement to the roof with horizontal cables on each floor connected to it by special amplifiers (repeaters). In some building the horizontal cables are thin, and the backbone thick.

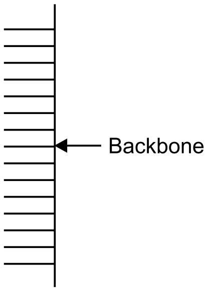


Fig. 5.12 Spine topology

3. *Tree topology:* The most general topology is the tree because a network with two paths between some pairs of stations would suffer from interference between the two signals.

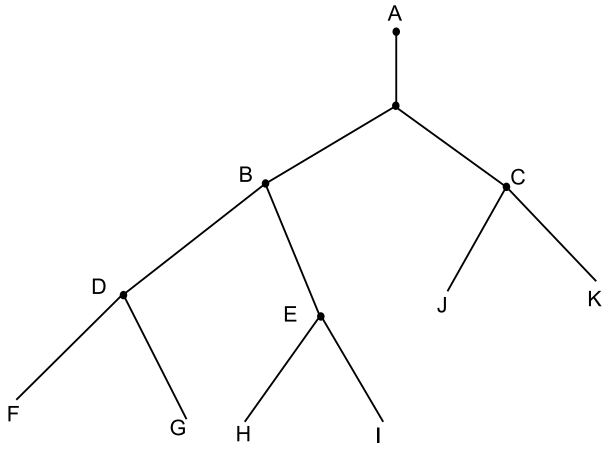


Fig. 5.13 Tree topology

4. *Segmented topology:* Even version of 802.3 has a maximum cable length per segment. To allow large networks, multiple cables can be connected by repeaters.

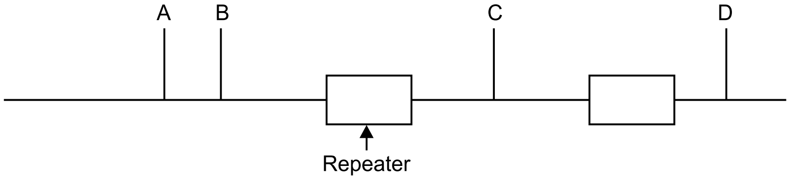


Fig. 5.14 Segmented topology

"A repeater is a physical layer device, which receives, amplifiers and retransmits signals in both directions."

As far as the S/W is concerned, a series of cable segment connected by repeater is no different than a single cable. A system may contain multiple cable segment and multiple repeaters, but no two transreceivers may be more than 2.5 km apart and no path between any two transreceivers may traverse more than four repeaters.

Connector Standards

In networking computers connect to many different types of terminals/devices. The wide variety of devices requires equally diverse connectors. Different types of connectors and their uses are described as follows.

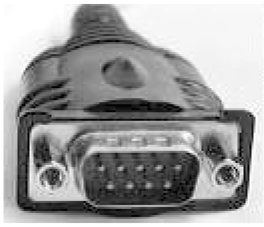


Fig. 5.15 DB-9

DB-9

The DB-9 connector is generally used for serial ports. Serial ports are used for devices that require a moderate speed from the interface. Serial ports support an 8-bit transfer but only one bit at a time.

The easiest way to picture this is to imagine a group of eight children waiting to go down a slide. Each child represents one bit sliding down in a single file line.

The most common peripheral that uses the DB-9 connector is the external modem.

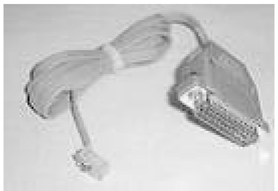


Fig. 5.16 DB-25

DB-25

The DB-25 connector is used for either a serial or parallel port. It is easy to tell the application of the connector by its type (male or female) on the back of the computer. Serial ports using a DB-25 are always male on the back of the PC, and parallel or printer ports are always female on the back of the PC.

If the connector is being used as a parallel interface, the speed will be faster than for a serial port.

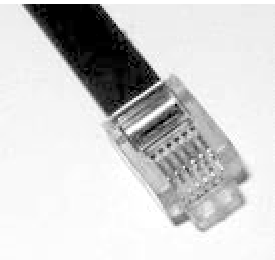


Fig. 5.17 RJ-11

RJ-11

The RJ-11 connector is used to connect a 2-pair wire to a receiving jack. The most common example of the RJ-11 connector is a telephone cord. The clear plastic connector on the end of the cord that you plug into the telephone set or into your modem is an RJ-11 connector.

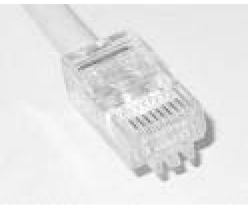


Fig. 5.18 RJ-45

RJ–45

The RJ-45 connector is the industry standard for Ethernet or Fast Ethernet networking. This 4-pair connector allows a short network cable, known as a patch cable, to attach the computer to the wall jack. This connector is found in 80% of networked businesses.

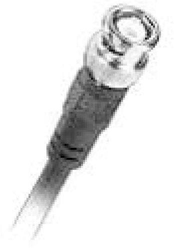


Fig. 5.19 BNC

BNC

The BNC (Bayonet Network Connector) connector looks like a barrel, which led to its nickname as the 'barrel connector'. BNC connectors are normally attached to a T connector at the work station. The T connector allows the PC to be attached to the network.

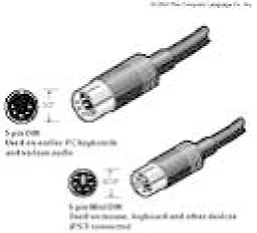

5 pin DIN
Used on earlier PC keyboards and Macintosh audio
6 pin Mini DIN
Used on mouse, keyboard and other devices (PS/2 connector)

Fig. 5.20 PS2/MINI-DIN

PS2/MINI-DIN

The PS2/Mini–DIN connector is most commonly used to connect keyboards and mouse to the back of the PC. The Mini-DIN is the most common type of peripheral connector used.

USB

USB is the fastest growing interface type at this time. The flexibility of the device's architecture is providing manufactures with a high-speed chainable port system that is easy to configure.

USB devices can be chained with the use of hubs, allowing up to 32 devices to be connected to one port. The transfer rate is also very good, with a maximum throughput of 4 Mbps.

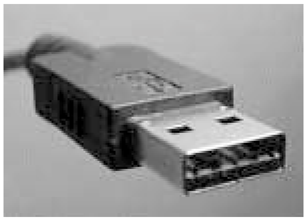


Fig. 5.21 USB

5.2.2 Network Interface Card (NIC)

A NIC is a hardware board or card that you put into an empty slot in the back of your client computer or server. NIC is the interface between the PC and the physical network connection. This card physically connects to the cable that links your network. As with any other type of adapter card NICs come in ISA, PCMCIA and PCI bus variaties. Fig. 5.22 shows typical Ethernet NIC. Since the NIC contains both BNC and RJ-45 connectors, it is called a combo card.

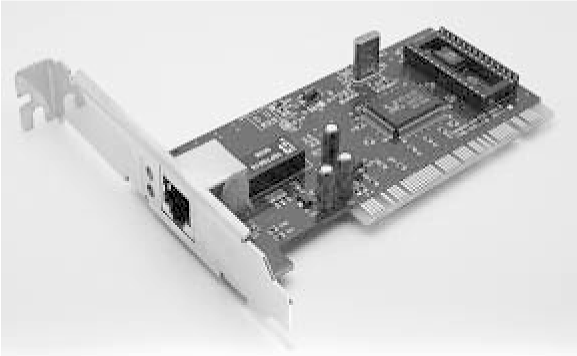


Fig. 5.22 Network interface card.

In addition to providing the physical connection to the networks, they also perform the following:

Prepare data

NIC prepare data so that it can transmit through the cable. The card translates data bit back and forth as they go from the computer to the cable and back again.

Address data

Each NIC has its own unique address that it imparts to the data stream. The card provides the data with an identifier. When it goes out on to the net and enables data seeking a particular computer to known where to exit the cable.

Control data flow

The card has RAM on it to help it, place the data so that it doesn't overwhelm the receiving computer on the cable.

Make (and agree on) the connection to another computer

Before it actually sends data, the NIC an electronic dialog with the other PC on the network that wants to communicate. They agree on thing like the maximum size of data groups to be sent. The total maximum size of data (amount), the time interval between data checks the amount of time that will elapse before confirmation that the data has arrived successfully and how much data each card hold before it overflows.

NIC is an especially useful place to implement IP sec. technology. This is the place where end station data is turned into useful security management information where data can be queued in order of priority before transport and where hardware acceleration can be used to the greatest advantage to help facilitate encryption.

An encrypted audio/video stream from a server to its clients provides a good example of the benefits of hardware acceleration. Users would experience much better network performance, if the stream were decrypted on an IP. (see enabled NIC).

Instead of via decryption software only hardware acceleration in the NIC can help to improve network performance by accelerating the many math cycle required by encryption and decryption algorithms by offloading the process onto a NIC problems are avoided.

Data transfers between the interfaces or nodes takes using these hardware address.

5.2.3 Bridges/Switches

Bridges are both hardware and software devices. They can be standalone devices. Separate boxes specifically designed for bridging applications, or they can be dedicated PCS with 2 NICs and bridging software.

Most servers software will automatically act as a bridge when a second NIC card is installed.

Bridges provides most economical means of interconnecting two or more LANs. Because bridges operates independently of network protocol, they are universal in nature.

On the negative side, bridge tend to-let extraneous network traffic, cross over one network to other. Bridges are store and forward devices. They receive a packet on the local segment, store it and wait for the remote segments to be clear before forwarding the packet.

There are two physical types of bridges.

1. Local Bridges

They are used where the network is being locally (talking physical location now) segmented. The two segment are physically close together *i.e.*, in same building same floor etc., only one bridge is then required.

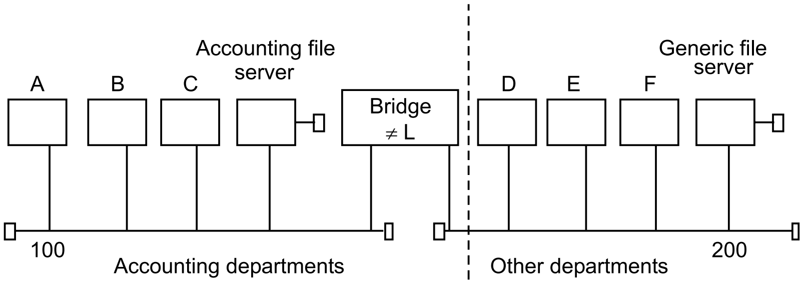


Fig. 5.23 Local bridge requirement

2. Remote Bridges

They are used in pairs and where the network is remotely segmented. These two segment are physically far apart like for different buildings, different floors, etc. 2 * half bridges are required, one at each segment.

The remote bridges are half of the normal bridge and may use several different communications media in between.

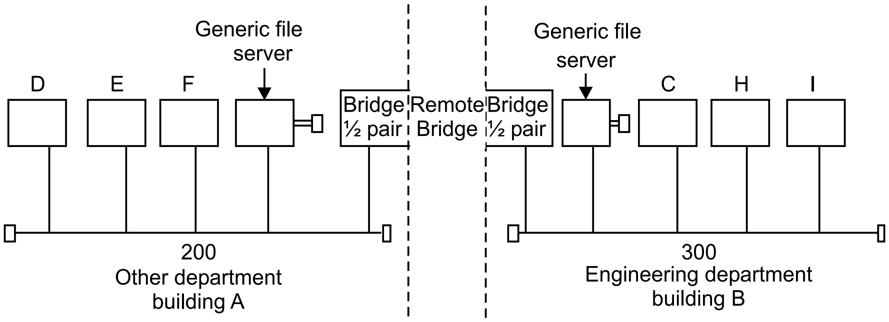


Fig. 5.24 Remote bridge connections

Purposes of bridges

Following are the purposes:

1. Isolate network by MAC addresses.
2. Manage network traffic by filtering packet.
3. Translate from one protocol to another.

1. Isolate network by MAC addresses

Let us consider that you have one segment called segment 100 with 50 users in several departments using this network segment.

Here one bridge is used to isolate the account departments and another bridge is used to isolate the engineering department.

The bridge will only allow to pass through that are not on the local segment. The bridge will first check its 'routing' table to see if the packet is on the local segment, if it is it will ignore the packet and not forward it to the remote segment

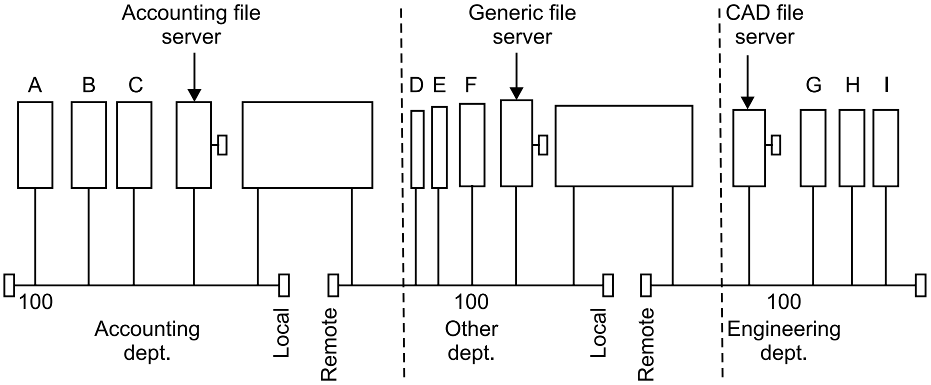


Fig. 5.25 Bridges connecting various departments

If client A sent a packet to the accounting file server, Bridge ≠ 1 will check its routing table, to see if the accounting file server is on the local port. If it is on the local port, Bridge ≠ 1 will not forward the packet to the other networks segment.

If client A sent a packet to the generic file server again bridge ≠ 1 will check its routing table to see if the generic file server is on the local port, if it is not then bridge ≠ 1 will forward the packet to the remote port.

2. Manage network traffic by filtering packets.

Bridges listen to the network traffic and build an image of the network on each side of the bridge. This image of the network indicates the location on each node and the bridge port that accesses it with this information a bridge can make a decision whether to forward the packet across the bridge, if the destination address is not on the same port or it can decide to not forward the packet if the destination is on the same ports.

This process of deciding whatever or not to forward a packet is termed filtering packets network traffic is managed by dividing which packets can pass through the bridge. The bridge filters packets.

3. Translate from one protocol to another

The MAC layer also contain the bus arbitration method used by the network, this can be CSMA/CD as used in Ethernet or token passing as used in token ring.

Bridges are aware of the BUS arbitration and special translation bridges can be used to translate between Ethernet and token ring.

There are three bridging methodologies used by bridges for connecting LANs.

Transparent Bridge

These are originally developed to support the connection of Ethernet networks. These bridges examine the MAC address of the frames to determine whether the packet is on the local segment or on the distant segment. Each bridge required to manually build the routing table. This manually building a routing table is called fixed/static routing.

Spanning Tree Protocol

These are developed to improve upon transparent bridging. The spanning tree protocol was developed to address the problems of loops in transparent bridging. The IEEE 802.1 D committee formed the STD. It converts a loop into a tree topology by disabling a bridge link.

It is a bridge to bridge communication where all bridges co-operate to form the overall bridge topology.

The spanning tree topology has algorithm which is dynamic and periodically checks every 1 to 4 sec to see if the bridge topology has changed.

Source Routing Bridges

Although transparent bridging can be used with token ring networks, IBM has promoted another bridging method called source routing.

With source routing, the bridge does not keep track of the route by which packets are sent. Unlike transparent bridges, source routing bridges allow redundant paths.

Reasons to use a bridge:

1. ***Security:*** Stops networks from forwarding sensitive data.
2. ***Bandwidth:*** Reduce traffic by segmentation.
3. ***Reliability:*** If one of the segment goes down, it does not take down the complete LAN.

4. ***Translation:*** Translate different data link protocols such as token ring to Ethernet.

A switch is similar to a bridge, with some important enhancements. First, a switch may have multiple ports, thus directing packets to several different segments, further partitioning and isolating network traffic in a way similar to a router. Fig. 5.26 shows an eight-port N-way switch, which can route packets from any input to any output. Some or all of an incoming packet is examined to make the routing decision, depending on the switching method that is used. One common method is called store and forward which stores the received packet before examining it to check for errors before retransmitting. Bad packets are not forwarded.

In addition, a switch typically has auto-sensing 10/100-Mbps ports and will adjust the speed of each port accordingly. Further more, a managed switch supports SNMP for further control over network traffic. Switches operate at layer 2 (Data Link) of the OSI model.

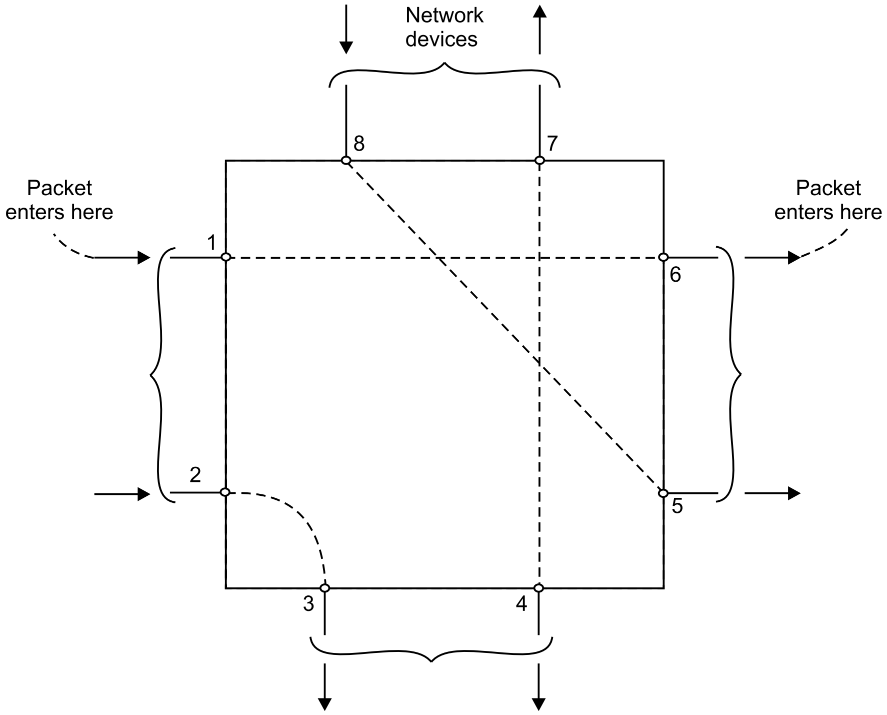


Fig. 5.26 One configuration in an eight port N-way switch

5.2.4 Routers

Routers are internetwork connectivity devices. An internetwork may consists of two or more physical connected independent network. These networks can be of different type.

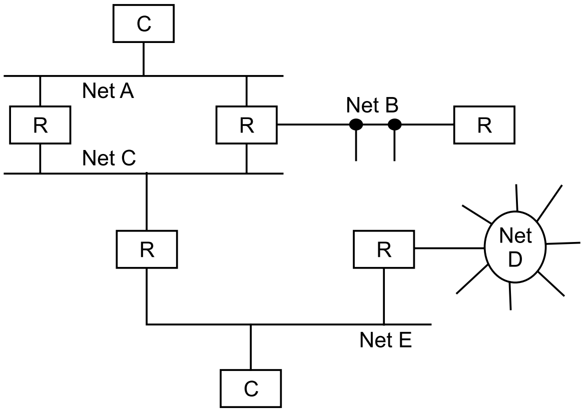


Fig. 5.27. Internet separated by router

For example, they can be ethernet and token ring network. Each network is logically separate and is assigned an address. Routers can use network address to assists efficient delivery of message.

Delivering packets according to logical network address is called routing. Routers performs routing. Routing is the process of finding a path from a source to every destination in the network.

Routers are intelligent. They can use algorithms to determine most efficient path for sending a packet to any given network.

Routers can also be used to divide large busy LANs into smaller segments. The protocols like IP, IPX and DDP are used to support routing functions.

Router are also employed to connect LAN to wide area network (WAN).

Routers are of two types.

1. *Static routers:* Static routers do not determine paths, but you need to specify them.
2. *Dynamic routers:* Dynamic routers have capacity to determine routes.

5.2.5 Concentrators

Concentrators acts as remote switching devices.

Suppose their are 100 subscriber in a locality who use their telephone number higher proportion of the time than the salesman use their desks. There is no need for 100 channels to connect them to their local switching office.

Here 20 channels could be used with some means of allocating a channel to a subscriber when be need it. This technique is called concentration. This is done in various way and variety of devices called concentrators.

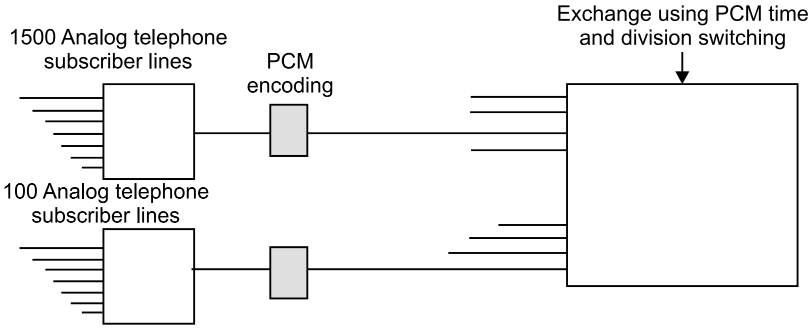

1500 Analog telephone subscriber lines
PCM encoding
Exchange using PCM time and division switching
100 Analog telephone subscriber lines

Fig. 5.28 Remote telephone concentrator designed for a PCM Line

Concentrator mechanism could be used in an apartment building and in a town street, if it were economical to connect a large number of subscribers to a smaller number of channels to the local central office, it is sometimes used in rural area, where subscribers are a long distance from their central office.

Concentrators are frequency used in data networks where their design is adjusted to the type of data traffic.

The design of concentrator depends on the type of signal, it is to concentrate. They read data into storage from lines with low utilization and then retransmits them over one or more lines with high utilization.

A concentrator for telephone lines may be an electromechanical device that scans bundle of lines searching for a free one.

It may be solid state circuits that concentrates PCM traffic to travel over a digital trunks. These can lower the cost of the local distribution networks.

5.2.6 Hubs

Hubs are special repeaters that overcome the electromechanical limitations of a media.

Hubs are wiring concentrators. They provide central attachment point for network cabling.

There are three types of hubs.

1. *Active hubs:* Active hubs can amplify and clean up the electronic signals. The process of cleaning up is called signal regeneration. Active hubs consists of electronic components.
2. *Passive hubs:* Passive hub is used to combine the signals from several network cable segment. This do not contain any electronic component and do not process the data signal in any way. All devices attached to passive hub receive all packets that pass through hub.

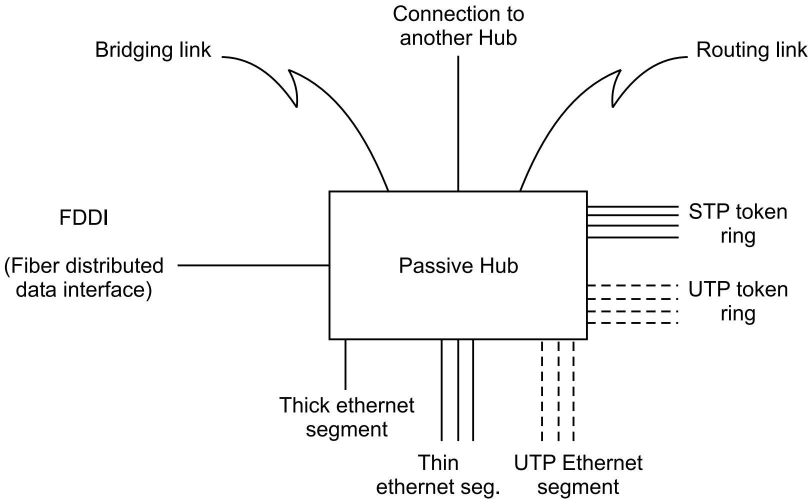


Fig. 5.29 Passive hub

3. *Switching hub:* It quickly routes the signal between ports of hub. These can be used in place of routers.

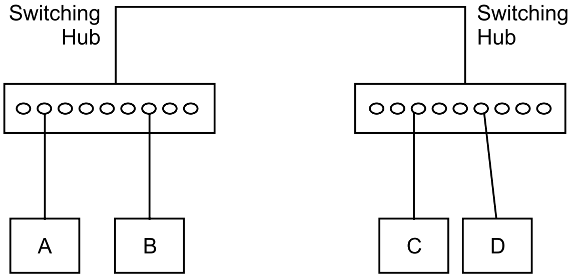


Fig. 5.30 Switching hubs

Consider Fig. 5.30. If A wanted to communicate with B, a dedicated 10 Mbps connection would be established between the two. If C wanted to communicate with D, another dedicated 10 Mbps connection would be established.

5.2.7 Repeaters

Repeaters are physical hardware devices that have a primary function to regenerate the electrical signal by,

- Reshaping the waveform.
- Amplifying the waveform.
- Retiming the signal.

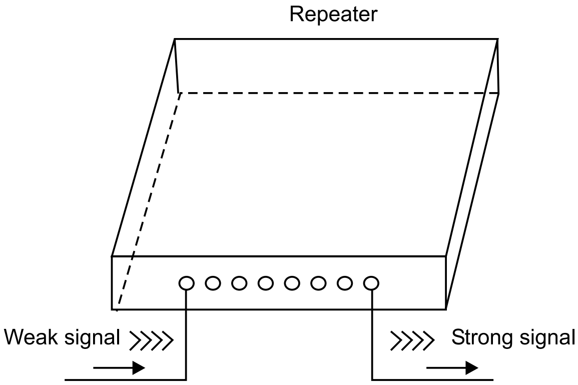


Fig. 5.31 Repeater

A repeater is a network device that repeats a signal from one port to another port. It does not filter or interpret.

This device is inexpensive and simple, by adding repeaters we can increase the length of network.

Purpose of a repeater

The purpose of a repeater is to extend the LAN segment beyond its physical limits as defined by the physical Layer's standards. (*e.g.* Ethernet is 500 m for 10 Base 5). A LAN segment is a logical path such as the logical bus used by all 802.3 Ethernet types. A LAN segment is given an identification number called segment number or network number to differentiate it from other segments.

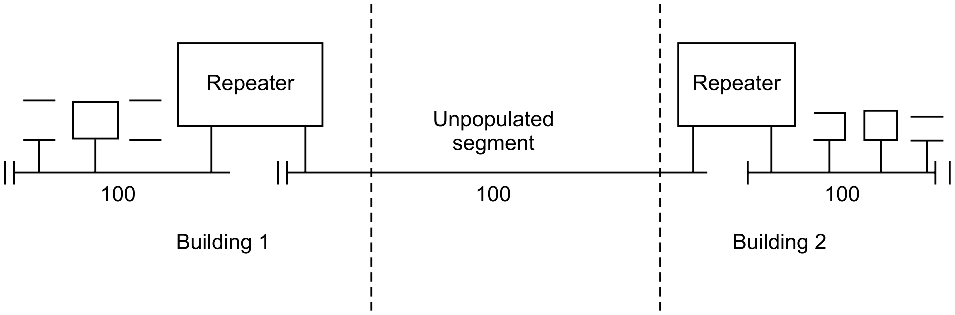


Fig. 5.32 Repeater connected to extend LAN

Typically, repeaters are used to extends the range of network cabling repeaters are also used to connect two physically close buildings together that are too far apart to just extend the segment. They can be used to connect floors of a building together that would surpass the maximum allowable segment length.

Note: For large extensions as shown in the figure 5.32, two Repeaters are required. For shorter extensions, only one Repeater may be required.

5.2.8 Gateways

Gateways solve the hard-to-reach problems that crop up when one type of network traffic needs to cross over or interact with a different type of network. You can safely assume that some form of gateway is available for virtually every unique network-to-network connectivity requirement that cannot be solved by router or a bridge.

A gateway is a server that appears to the client as if it were an origin server. It acts on behalf of other servers that may not be able to communicate directly with a client.

One definition of a gateway is the hardware/software device that is used to interconnect LANs and WANs with mainframe computers such as DEC net and IBM's SNA. Often the router that is used to connect a LAN to the Internet will be called a gateway. It will have added capability to direct and filter higher layer protocols (layer 4 and up) to specific devices such as web servers, ftp servers and e-mail servers.

Gateway's OSI operating layer

A Gateway operates at the transport layer and above which is shown in figure 5.33 Typically translating each source layer protocol into the appropriate destination layer, protocol, a mainframe gateway may translate all OSI model layers.

For, example, IBM's SNA (System Network Architecture) does not readily conform to the OSI model and requires a gateway to translate between the two architectures.

OSI Model	*SNA*
Application	Transaction Services
Presentation	Presentation Services
Session	Data Flow Control
Transport	Transmission Control
Network	Path Control
Data Link	Data Link Control
Physical	Physical

Fig. 5.33 OSI Model and SNA

Gateway's segment to segment characteristics

There can be major differences between 'local' and 'distance' segments. As can be seen from the Fig. 5.33, the two networks appear as if they are from other planets. Mainframes are based on a central number crunching CPU with terminals connected. The central CPU controls all information displayed on the terminals. LANs consist of distributed CPUs that share data and files. This leads to a unique problem in connecting the two architectures that requires a gateway.

Gateway addressing

The gateway addressing depends on which OSI layers are translated. It could be all layers.

5.2.9 ISDN

An international standards being developed for digital networks that will be able to accomodate voice, data, video and various other types of traffic.

ISDN (Integrated Sevices Digital Networks) is defined by the standardization of user interfaces and is implemented as a set of digital switches and paths supporting a broad range of traffic types and providing value added processing services.

There are two generations of ISDN.

1. *Narrowband ISDN:* Narrowband ISDN is based on the use of a 64 Kbps channels as the basic unit of switching and has a circuit switching orientation.
2. *Broadband ISDN:* Broadband ISDN, supports very high data rates (100 Mbps) and has a packet switching orientation.

A Conceptual View of Isdn

ISDN is a massive undertaking in many ways, and it is difficult to give a perfect description of it. The concept of ISDN is best introduced by considering it from several different view points.

- Principles of ISDN
- Evolution of the ISDN
- The user interface
- Objectives
- Benefits
- Services
- Architecture

Principles of ISDN

Standards for ISDN have been defined by ITU-T, which states the principles of ISDN.

1. **Support for voice and non-voice applications using a limited set of standardised facilities.**

   The ISDN supports a variety of services related to voice communications (telephone calls) and non-voice communications (digital data exchange). These services are to be provided in conformation with standards that specify a small no. of interfaces and data transmission facility.

2. **Supports for switched and non-switched applications.** The ISDN supports both circuit switching and packet switching. Also it performs non-switched services in the form of dedicated lines.
3. ***Reliance on 64 Kbps connections.*** **I**SDN provides circuit switching and packet switching connections at 64 Kbps. This is the fundamental building block of ISDN. This rate was chosen and it was the standard rate for digitized voice.
4. ***Intelligence in the network:*** An ISDN is expected to be able to provide sophisticated services beyond the simple setup or a circuit switched call.
5. ***Layered protocol architecture:*** The protocols for user access to ISDN exhibit a layered architecture and can be mapped into the OSI model. This has a number of advantages.
   - Standards already developed for OSI-related applications may be used on ISDN.
   - New ISDN-related standards can be based on existing standards, reducing the cost of new implementations.
   - Standards can be developed and implemented independently for various layers and for various functions within a layer.
6. ***Variety of configurations:*** More than one physical configuration is possible for implementing ISDN. This allows for differences in national policy in the state of technology and in the needs and existing equipment of the customer base.

Evolution of ISDN

ISDN evolves from and with the integrated digital network (IDN). The evolution of the IDN has been driven by the need to provide econmic voice communications. The resulting network, however, is also well suited to meet the growing variety of digital data service needs.

Evolution from telephone IDNs.

The intent is that the ISDN evolve from the existing telephone networks. Two conclusions can be drawn from this point.

1. The IDN technology developed for and evolving within existing telephone networks forms the foundation for the services to be provided by ISDN.
2. Although other facilities, such as third party (not the telephone provider) packet-switched networks and satellite links, will play a role in ISDN, the telephone networks will have the dominant role.

The overwhelming prevalence of telephone networks dictates that these networks form the basis for ISDN.

Transition of one or more decades

The evolution to ISDN will be a slow process. This is true of any migration of a complex application or set of applications from one technical base to a newer one. The introduction of ISDN services will be done in the context of existing digital facilities and existing services. There will be a period of coexistence in which connections and perhaps protocol conversion will be needed between alternative facilities and/or services.

Use of existing networks

This point is simply an elaboration of point 2 above. For example, ISDN will provide a packet-switched service. For the time being, the interface to that service will be X.25. With the introduction of fast packet-switching and more sophisticated virtual call control, there may need to be a new interface in the future.

Interim user-network arrangements

The lack of digital subscriber lines might delay introduction of digital services, particularly in developing countries. With the use of modems and other equipment, existing analog facilities can support at least some ISDN services.

Connections at other than 64 Kbps

The 64 Kbps data rate was chosen as the basic channel for circuit switching. With improvements in voice digitizing technology, this rate is unnecessarily high. On the other hand, this rate is too low for many digital data applications. Thus, other data rates will be needed.

The User Interface

The given figure 5.34 shows a conceptual view of the ISDN from a user or customer point of view. The user has access to the ISDN by means of a local interface to a digital “pipe” of a certain bit rate. Pipes of various sizes will be available to satisfy differing needs. For example, a residential customer may require only sufficient capacity to handle a telephone and a personal computer. An office will typically wish to connect to the ISDN via an on-premise digital PBX or LAN and will require a much higher capacity pipe.

More than one size of pipe will be needed is emphasized. At the low end of demand would be a single terminal (*e.g.*, a residential telephone) or multiple terminals in some sort of multidrop arrangement (*e.g.*, a residential telephone, personal computer, and alarm system). Offices are more likely to contain a network of devices attached to a LAN or PBX, with an attachment from that network acting as a gateway to the ISDN.

At any given point in time, the pipe to the user’s premises has a fixed capacity, but the traffic on the pipe may be a variable mix up to the capacity limit. Thus, a user may access circuit-switched and packet-switched services, as well as other services, in a dynamic mix of signal types and bit rates. The ISDN will require rather complex control signals to instruct it how to sort out the time-multiplexed data and provide the required services. These control signals will also be multiplexed onto the same digital pipe.

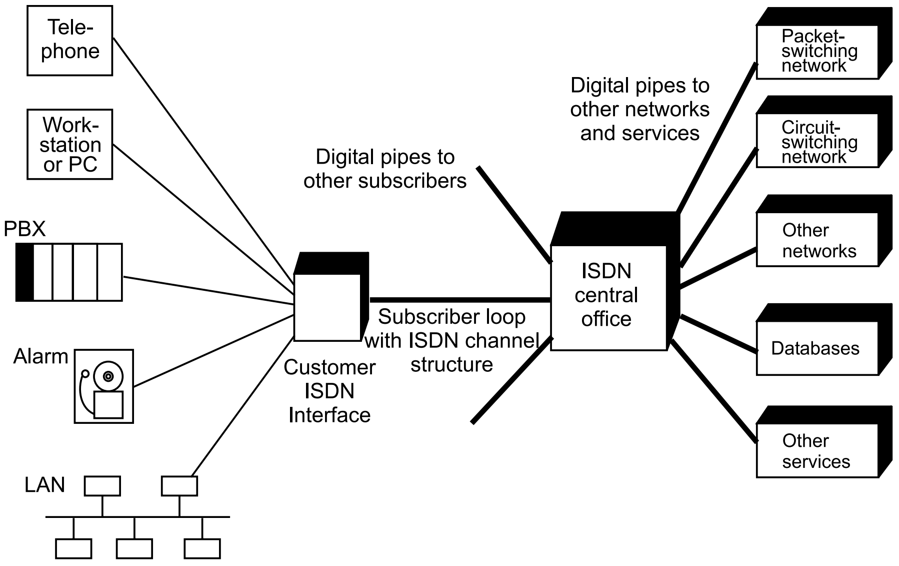

Tele-phone
Work-station or PC
PBX
Alarm
LAN
Customer ISDN Interface
Subscriber loop with ISDN channel structure
Digital pipes to other subscribers
ISDN central office
Digital pipes to other networks and services
Packet-switching network
Circuit-switching network
Other networks
Databases
Other services

Fig. 5.34. Conceptual view of ISDN connection features

An important aspect of the interface is that the user may, at any time, employ less than the maximum capacity of the pipe and will be charged according to the capacity used rather than "connect time".

OBJECTIVES

Activities currently under way are leading to the development of a worldwide ISDN. This effort involves national governments, data processing and communication companies, standards organizations, and others. Certain common objectives are, by and large, shared by this disparate group. The key objectives are as follows:

1. Standardization.
2. Transparency.
3. Separation of competitive functions.
4. Leased and switched services.
5. Cost-related tariffs.
6. Smooth migration.
7. Multiplexed support.

1. Standardization

Standardization is essential to the success of ISDN standards will provide for universal access to the network. ISDN standard equipment can be moved from one location to another, indeed from one country to another, and be plugged into the network.

2. Transparency

It is also important that the digital transmission service have the property of transparency that is, the service is independent of, and does not affect, the content of the user data to be transmitted. This permits users to develop applications and protocols with the confidence that they will not be affected by the underlying ISDN.

3. Separation of competitive functions

The ISDN must be defined in a way that does not preclude the separation of competitive functions from the basic digital transmission services. It must be possible to separate out functions that could be provided competitively as opposed to those that are fundamental part of the ISDN.

4. Leased and switched services

The ISDN should provide both leased and switched services. This will give the user the greatest range of options in configuring network services and allow the user to optimize on the basis of cost and performance.

5. Cost-related tariffs

The price for ISDN service should be related to cost and independent of the type of data being carried. Such a cost-related tariff will assure that one type of service is not in the position of subsidizing others. Price distinctions should be related to the cost of providing specific performance and functional characteristics of a service.

6. Smooth migration

For an extended period of time, the evolving ISDN must coexist with existing equipment and services. To provide for a smooth migration to ISDN, ISDN interfaces should evolve from existing interfaces, and interworking arrangements must be designed. Specific capabilities that will be needed include adapter equipment that allows pre-ISDN terminal through a mixed ISDN/non-ISDN network complex, and protocol converters to allow interoperation of ISDN services and non-ISDN services.

7. Multiplexed support

Multiplexed support must be provided to accommodate user-owned PBX and local area network (LAN) equipment.

Benefits

The principal benefits of ISDN to the customer can be expressed in terms of cost savings and flexibility. The integration of voice and a variety of data on a single transport system means that the user does not have to buy multiple services to meet multiple needs. The efficiencies and economies of scale of an integrated network allow these services to be offered at lower cost than if they were provided separately.

Network providers, on a larger scale but in a similar way, profit from the advantages of competition, including the areas of digital switches and digital transmission equipment.

Manufacturers can focus research and development an technical applications and be assured that a broad potential demand exists.

Finally, enhanced service providers of, for instance, information-retrival or transaction-based service, will benefit from simplified user access.

Of course, any technical innovation comes with penalities as well as benefits.

- The cost of migration.
- It will retard technical innovation.

Services

The ISDN will provide a variety of services, supporting existing voice and data applications as well as providing for applications now being developed. Some of the important applications are as follows.

Facsimile

Service for the transmission and re-production of graphics and hand-written and printed material. This type of service has been available for many years but has suffered from a lack of standardization and the limitations of the analog telephone network. Digital facsimile standards are now available and can be used to transmit a page of data at 64 Kbps in 5 seconds.

Teletex

Service that enables subscriber terminals to exchange correspondence. Communicating terminals are used to prepare, edit, transmit, and print messages. Transmission is at a rate of one page in 2 seconds at 9.6 Kbps.

Videotex

An interactive information retrieval service. A page of data can be transmitted in 1 second at 9.6 Kbps.

These services fall into the broad categories of voice, digital data, text, and image. Most of these services can be provided with a transmission capacity of 64 Kbps or less. Some services require considerably higher data rates and may be provided by high speed facilities outside the ISDN.

Architecture

Fig. 5.35 is an architectural view of ISDN. The ISDN will support a completely new physical connector for users, a digital subscriber line, and a variety of transmission services.

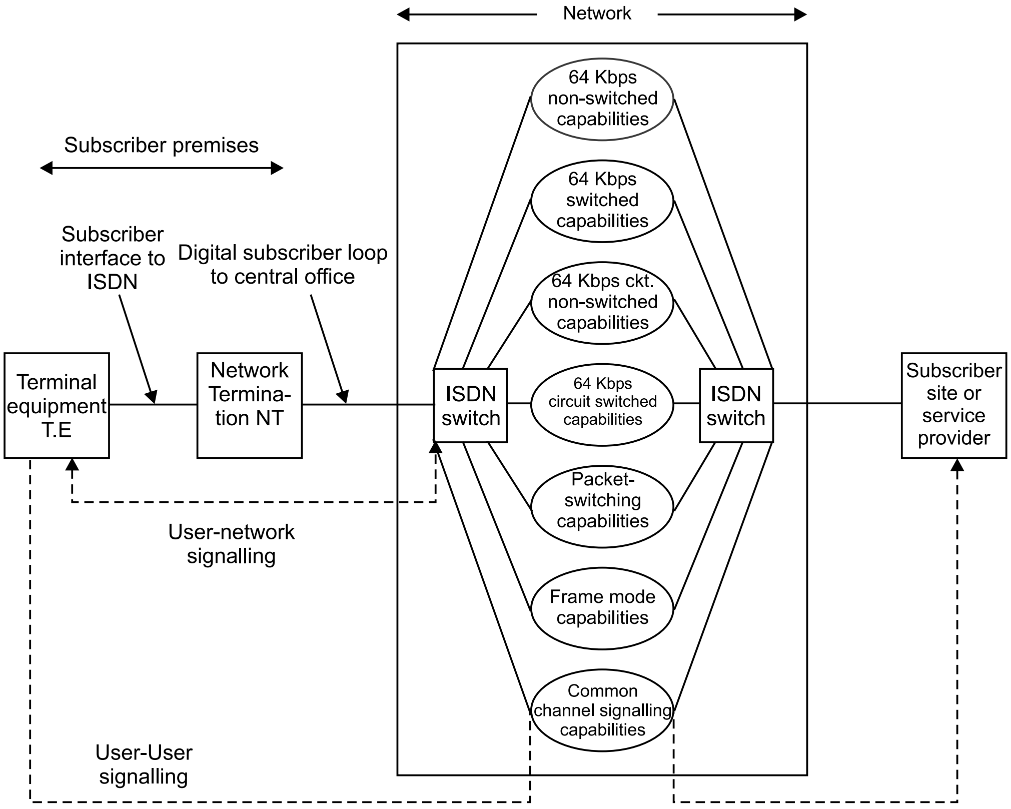

Network
Subscriber premises
Subscriber interface to ISDN
Digital subscriber loop to central office
Terminal equipment T.E
Network Termina-tion NT
ISDN switch
ISDN switch
Subscriber site or service provider
64 Kbps non-switched capabilities
64 Kbps switched capabilities
64 Kbps ckt. non-switched capabilities
64 Kbps circuit switched capabilities
Packet-switching capabilities
Frame mode capabilities
Common channel signalling capabilities
User-network signalling
User-User signalling

Fig. 5.35. ISDN architecture

The common physical interface provides a standardized means of attaching to the network. The same interface should be usable for telephone, personal computer, and videotex terminal. Protocols are required to define the exchange of control information between user device and the network. The interface supports a basic service consisting of three time-multiplexed channels, two at 64 Kbps and one at 16 Kbps. Also there is a primary service that provides multiple 64-Kbps channels.

For both services, an interface is defined between the customer's equipment, referred as terminal equipment (TE), and a device on the customer's premises, known as a network termination (NT). The NT forms the boundary between the customer and the network.

The subscriber line is the physical signal path from the subscriber's NT to the ISDN central office. This line must support full duplex digital transmission for both basic and primary data rates.

The ISDN central office connects the numerous subscriber lines to the digital network. This provides access to a variety of lower-layer transmission facilities along with following:

1. ***Circuit-Switched Capabilities:*** Operating at 64 Kbps, this is the same facility provided by other digital switched telecommunications networks.
2. ***Non-switched Capabilities:*** A non-switched capability at a higher data rate is to be provided by broadband ISDN and will be in the nature of a permanent virtual circuit for asynchronous transfer mode (ATM) transmission. It offers 64 Kbps dedicated link.
3. ***Switched Capabilities:*** This refers to high speed (> 64 Kbps) switched connections using ATM as part of broadband ISDN.
4. ***Packet-Switched Capabilities:*** This facility resembles packet-switched service provided by other data networks.
5. ***Frame-Mode Capabilities:*** A service that supports frame relay.
6. ***Common-Channel Signalling Capabilities.*** These are used to control the network and provide call management.

THE I-SERIES RECOMMENDATIONS

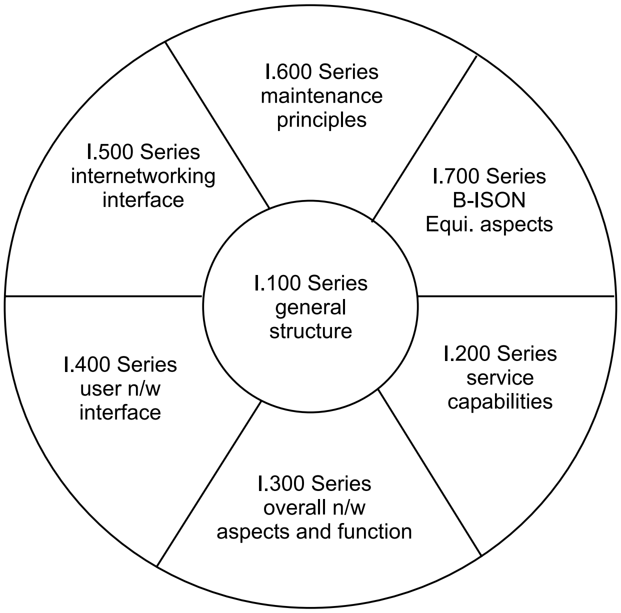


Fig. 5.36. Structure of the I-series recommendations

The development of ISDN is governed by a set of recommendations, called the I-series recommendations.

The I-series recommendations are broken up into seven main groupings, labelled as I.100 to I.600.

1. *I.100 Series : General Concept*: All I.100 series serves as a general introduction to ISDN. I.120 provides an overall description of ISDN and the expected evolution of ISDNs. I.130 introduces terminology and concepts that are used in the I.200 series to specify services.
2. *I.200 Series : Service Capabilities*: The I.200 series is in a sense the most important part of the ITU-T ISDN recommendations. Here the services to be provided to users are specified. In the ISDN glossary (I.112), the term "service" is defined as, that which is offered by an administration or recognised private operating agency (RPOA) to its customers in order to satisfy a specific telecommunication requirement.

   The term "service" has come to have a very specific meanings in ITU–T, *i.e.*, somewhat different from the use of that term in an OSI context. For ITU–T a standardised service is characterised by

   - Complete, guaranteed end-to-end compatibility.
   - ITU-T standardised terminals, including procedures.
   - Charging and accounting rules.

   There are *three* fully standardised ITU–T services. *i.e.* Telegraphy, Telephony and data.

   There are *four* additional telematic services in the process of being standardised. Teletex, facsimile, videotex and message handling ensure high-quality international telecommunications for the end user, regardless of the make of the terminal equipment and the type of network used nationally to support the service.
3. *I.300 Series : Network Aspects*: I.300 series focuses in terms of how the network goes about providing those services on the network.
4. *I.400 Series : User-Network Interfaces*: I.400 series deals with the interface between the user and the network, having following three points, *i.e.* physical configuration, transmission rate and protocol specifications.
5. *I.500 Series : Internetwork Interfaces*: ISDN supports services that are also provided on older circuit-switched and packet-switched networks. Thus, it is necessary to provide internetworking between an ISDN and other types of networks.
6. *I.600 Series : Maintenance Principles*: This series provide guidance for maintenance of the ISDN subscriber installation, the network portion of the ISDN basic access, primary access, and higher data rate services.
7. *I.700 Series : B-ISDN Equipment Aspects*: This series was first introduced in 1996. It covers functionality and characteristics of ATM equipment and various management aspects.

ISDN CHANNELS

The digital pipe between the central office and the ISDN users is used to carry a number of communication channels. The transmission structure of any access link is constructed from the following channels.

- B channels : 64 Kbps.
- D channels : 16 or 64 Kbps.
- H channels : 384 (H_0), 1536 (H_{11}), and 1920 (H_{12}) Kbps.

B Channel

The B channel is the basic user channel. It can be used to carry digital data, PCM-encoded digital voice, or a mixture of lower rate traffic including digital data and digitized voice encoded at a fraction of 64 Kbps. In the case of mixed traffic, all traffic must be destined for the same end point. Four kinds of connections can be setup over a B channel.

1. ***Circuit-Switched:*** This is equivalent to switched digital service available today. The user places a call, and a circuit-switched connection is established with another network user, An interesting feature is that call-establishment dialogue does not take place over the B channel, but is done over the D.
2. ***Packet-Switched:*** The user is connected to a packet-switching node, and data are exchanged with other users via X.25.
3. ***Frame Mode:*** The user is connected to a frame relay node, and data are exchanged with other users via LAPF.
4. ***Semipermanent:*** This is a connection to another user setup by prior arrangement, and not requiring a call-establishment protocol. This is equivalent to a leased line.

D Channel

The D channel serves two purposes.

1. It carries signalling information to control circuit-switched calls on associated B channels at the user interface.
2. The D channel may be used for packet-switching or low-speed (*e.g.*, 100 bps), telemetry at time when no signaling information is waiting.

The functions of ISDN D channels are signalling Basic, Enhanced low speed data. Videotex, Teletex, Terminal, Telemetry.

H Channel

H channels are provided for user information at higher bit rates. The user may employ such a channel as a high speed trunk. Examples of applications include fast facsimile, video, high-speed data, high-quality audio, and multiple information streams at lower data rates.

These channels types are grouped into transmission structure that are offered as a package to the user. The best defined structures at this time are the basic channel structure (basic access) and the primary channel structure (primary access) which is shown in Fig. 5.37.

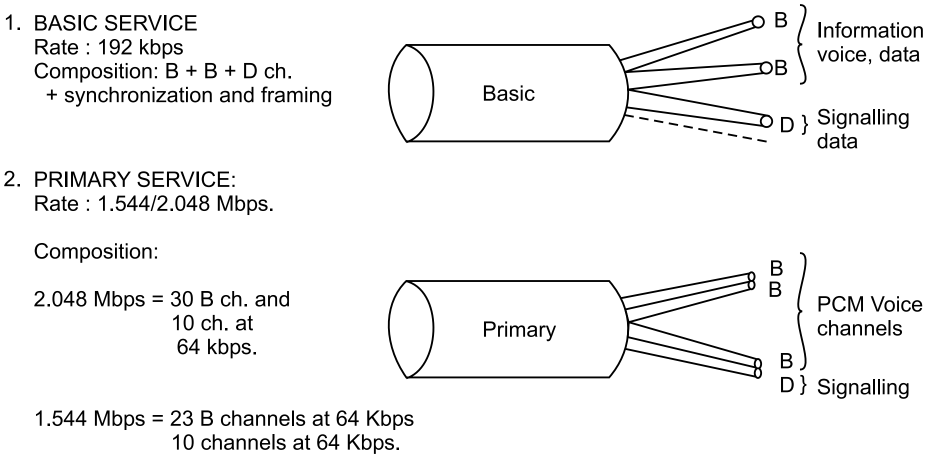


Fig. 5.37. ISDN channel structure

BASIC ACCESS

Basic access consists of two full duplex 64 Kbps B channels and a full duplex 16 Kbps D channel. The total bit rate, by simple arithmetic, is 144 Kbps. However, framing, synchronization and other overhead bits bring the total bit rate on a basic access link to 192 Kbps.

PRIMARY ACCESS

Primary access is intended for users with greater capacity requirements, such as offices with a digital PBX or a local network. Because of differences in the digital transmission hierarchies used in different countries, it was not possible to get agreement on a single data rate.

ISDN USER ACCESS

To define the requirements for ISDN user access, an understanding of the anticipated configuration of user premises equipment and of the necessary standard interfaces is critical.

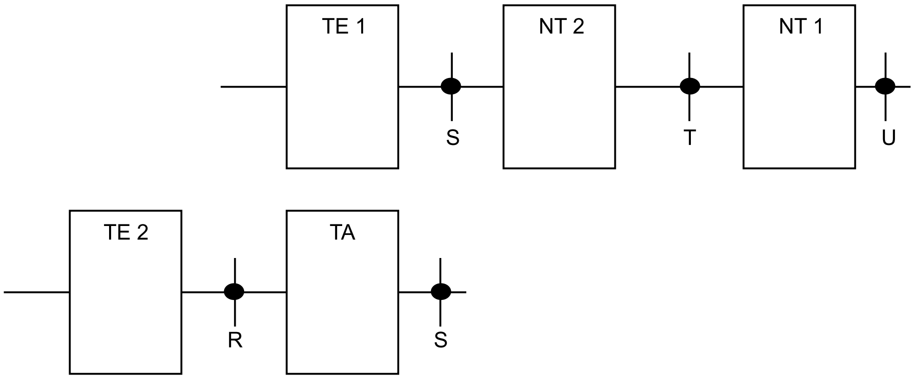


Fig. 5.38. ISDN reference points and functional groupings

The Fig. 5.38 shows CCITT approach to this task, using,

- ***Functional Grouping:*** It is a certain finite arrangement of physical equipment or combinations of equipment.
- ***Reference Points:*** These are the conceptual points used to separate groups of functions.

The architecture on the subscriber's premises is broken up functionally into groupings separated by reference points. This separation permits interface standard to be developed at each reference point. This effectively organises the standard work and provides guidance to the equipment providers. Once stable interface standards exist, technical improvement on either side of an interface can be made without impacting adjacent functional groupings.

Finally, with stable interface, the subscriber is free to procure equipment from different suppliers for the various functional groupings so long as the equipment conforms to the relevant interface standards.

BROADBAND ISDN

In 1988, as part of its I-series of recommendations on ISDN, CCITT issued the first two recommendations relating to broadband ISDN.

CCITT modestly defines B-ISDN as, "a service requiring transmission channels capable of supporting rates greater than the primary rate".

With B–ISDN services especially video services, requiring data rates in excess of those that can be delivered by ISDN will became available with constrating this B-ISDN with ISDN then original concept is now being reffered to as narrow band ISDN.

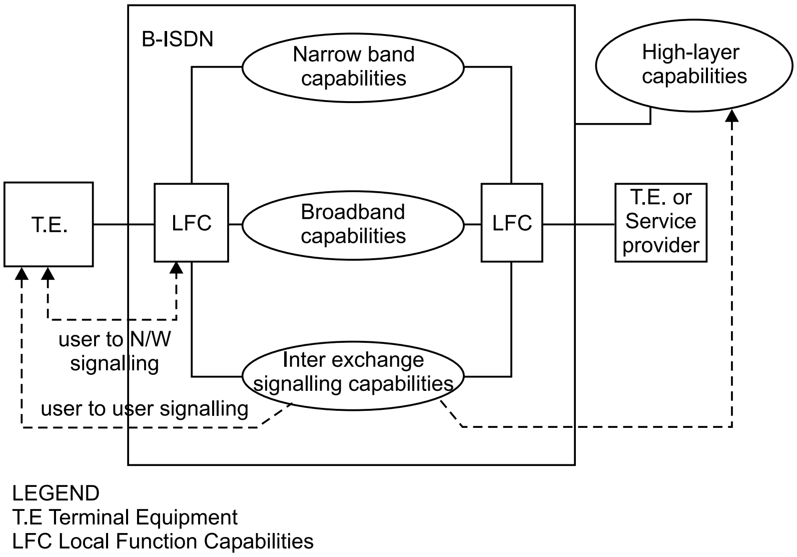


Fig. 5.39 B-ISDN architecture

B-ISDN differs from a narrow band ISDN in a number of ways. To meet the requirement for high resolution video, an upper channel rate of approximately 150 Mbps is needed. To simultaneously support one or more interactive and distributive services a total subscriber line rate of about 600 Mbps is needed.

The introduction of B-ISDN depends on the pace of introduction of fiber subscriber loops. Internal to the network, there is the issue of the switching technique to be used. The switching facility has to be capable of handling a wide range of different bit rates and traffic parameters (*e.g.* burstness).

The Fig. 5.39 shows the functional architecture B-ISDN. As with narrow band ISDN control of B-ISDN is based on common channel signalling within the network as SS7, enhanced to support the expanded capabilities of a higher speed network is used.

B-ISDN must support all of the 64-Kbps transmission services, both circuit switching and packet switching, that is supported by narrowband ISDN.

Broadband capabilities are provided for higher data rate transmission services. The broadband functional groups are equivalent to the functional groups defined for narrow band and interfaces at the R reference point may or may not have broadband capabilities.

The datarates available to B-ISDN subseries defines three new transmission services.

1. A full duplex 155.52 Mbps service.
2. Asymmetrical providing transtion from the subscriber to the network at 155.52 Mbps and in the other direction at 622.08 Mbps.
3. A full duplex, 622.08 Mbps service.

The protocol reference model of B-ISDN makes reference to three separate planes.

User Plane

Provides user information transfer alongwith associated controls. (*e.g.* Flow control, Error control).

Control Plane

Performs call control and connection control functions.

Management Plane

It performs management functions related to a system as a whole.

QUESTIONNARIES

1. What is mean by Layered Approach in computer networking?
2. Explain interfaces in computer networking and explain connection oriented and connectionless services.
3. Explain basic concept of protocols and give functions of protocols.
4. What is X.25? Explain types of virtual circuits.
5. Explain X.25 packet-switched network.

6. Explain the services provided by X.25 protocol.
7. Explain Intranet and Extranet in networking.
8. Draw and explain five alternatives of LAN cabling.
9. Explain different LAN Topology.
10. Draw and explain ISDN architecture and what are advantages of ISDN.
11. Discuss the different types of channels that are used to construct the transmission structure of the access link of the ISDN.
12. Explain I-series recommendations related to ISDN.
13. Explain the conceptual view of ISDN user-network interface.
14. Draw and explain architecture of B-ISDN.
15. Write short note on
    - NIC (Network Interface Card)
    - Bridges
    - Hub
    - Gateways
    - Principles of B-ISDN
    - Reference points and functional groupings.

Chapter 6

PHYSICAL LAYER

INTRODUCTION

In this chapter we will look at the lowest layer depicted in the hierarchy of the OSI reference model. We will begin with a theoretical analysis of data transmission.

Then this chapter covers transmission media, both guided (Copper Wire and Fiber Optics) and unguided (wireless). This material will provide background information on the key transmission technologies used in modern networks. Also in this chapter we will discuss about modem.

6.1. PHYSICAL LAYER CHARACTERISTICS

The physical layer is the lowest of the seven layers and establishes the physical connection and interfaces to the transmission medium. The characteristic of this layer are independent of the media, which could be co-axial cable, Twisted copper wire, Fiber optic cable and many other cable types.

The physical layer has four important characteristics.

Mechanical

It defines the physical attributes of the connector, such as number of pins, shape and dimension of the connecting block and so on.

Electrical

Relates to the representation of bits. (In terms of voltage levels) and the data transmission rate of bits. Also it specifies whether the connection is balanced or unbalanced.

Functional

It specifies the function performed by individual circuit of the physical interfaces between a system and the transmission medium. Such as a function, data control timing and earth.

Procedural

It specifies the sequence of events by which bit streams are exchanged across the physical medium.

There is also a specification for the physical layer relay which acts as a bridge between different transmission medium, such as copper telephone wire to co-axial cable, to fiber optic circuit and so on.

6.2 THE THEORETICAL BASIS FOR DATA COMMUNICATION

Information can be transmitted on wires by varying some physical property such as voltage or current. By representing the value of this voltage or current as a signal valued Function of time, F(t), we can model the behaviour of the signal and analysis it mathematically.

6.2.1 Fourier Analysis

In the early 19th century, the French mathematician **Jean-Baptiste Fourier** proved that any reasonably behaved periodic function, g(t), with period T can be constructed by summing a (possibly infinite) number of sines and cosines.

$$g(t) \;=\; \frac{1}{2}c + \sum_{n=1}^{\infty} a_n \sin(2\pi n f t) + \sum_{n=1}^{\infty} b_n \cos(2\pi n f t) \qquad \text{...(1)}$$

where,

$f = \frac{1}{T}$ is the functional frequency

a_n = sine amplitude of the n^{th} harmonic (term)

b_n = cosine amplitude

Such a decomposition is called a **Fourier series**.

From the Fourier series the function can be reconstructed *i.e.*, if the period T is known and amplitude is given the original time function can be found by performing the sum of equation (1).

Here data signal that has a finite duration which, all of the terms can be handled by just imaging that it repeats the entire pattern over and over forever.

i.e., the interval from T to 2T is the same as 0 to T.

→ The amplitude can be computed for any given g(t) by multiplying both side of equation (1) by sin (2πkft).

$$\therefore\ g(t)\sin(2\pi kft) = \frac{1}{2}c\sin(2\pi kft) + \sum_{n=1}^{\infty} a_n \sin(2\pi nft).\sin(2\pi kft) + \sum_{n=1}^{\infty} b_n \cos(2\pi nft).\cos(2\pi kft) \qquad \text{...(2)}$$

Integrating equation (2) from 0 to T.

$$\therefore \int_0^T g(t)\sin(2\pi kft)\,\delta t = \int_0^T \frac{c}{2}\sin(2\pi kft)\,\delta t + \int_0^T \sum_{n=1}^{\infty} a_n \sin(2\pi nft).\ \sin\ (2\pi\ kft)\ \delta t$$

$$+\int_0^T \sum_{n=1}^{\infty} b_n \cos(2\pi n f\ t).\sin(2\pi k f\ t)\ \delta t \qquad ...(3)$$

Suppose $2\pi f = \omega$ and $k = n$, equation (3) becomes.

$$\int_0^T g(t)\sin\omega_n t\ \delta t = \int_0^T \frac{c}{2}\sin\omega_n t\ \delta t + \sum_{n=1}^{\infty} a_n \int_0^T \sin\ \omega_n t.\sin\omega_n t\ \delta t + \sum_{n=1}^{\infty} b_n \int_0^T \cos\ \omega_n t.\sin\omega_n t\ \delta t$$

$$\int_0^T g(t)\sin\omega_n t\ \delta t = \frac{c}{2}\left[\frac{-\cos\omega_n t}{\omega_n}\right]_0^T + \overset{\xrightarrow{2^{nd}\text{ term}}}{\sum_{n=1}^{\infty} a_n \int_0^T \frac{1-\cos 2\omega_n t}{2}\ \delta t} + \underset{\xrightarrow[3^{nd}\text{ term}]{}}{\sum_{n=1}^{\infty} b_n \int_0^T \cos\omega_n t.\sin\omega_n t\ \delta t} \qquad ...(4)$$

$$\xrightarrow[1^{st}\text{ term}]{}$$

1st term:

$$= \frac{c}{2}\left[\frac{-\cos\ \omega_n t}{\omega_n}\right]_0^T$$

$$= \frac{-c}{2\omega_n}[0] = 0$$

2nd term:

$$= \sum_{n=1}^{\infty} a_n \int_0^T \frac{1-\cos 2\omega_n t}{2}\ \delta t$$

$$= \sum_{n=1}^{\infty} \frac{a_n}{2}\left[t - \frac{\sin\ 2\omega_n t}{2\omega_n}\right]_0^T$$

$$= \sum_{n=1}^{\infty} \frac{a_n}{2}[T - 0]$$

$$= \sum_{n=1}^{\infty} \frac{a_n}{2}\ T$$

3rd term:

$$= \sum_{n=1}^{\infty} b_n \int_0^T \cos\omega_n t.\sin\omega_n t\ \delta t$$

$$= \sum_{n=1}^{\infty} b_n \left[\frac{\sin 2\omega_n t - \sin 2\omega_n t.0}{2\omega_n}\right]_0^T$$

$$= \frac{-b_n}{2\omega_n}[0] = 0$$

Substituting the values of these three terms in equation (4),

$$\therefore \qquad \int_0^T g(t)\,\sin \omega_n t\ \delta t \;=\; 0 + \sum_{n=1}^{\infty} \frac{a_n}{2} T + 0$$

$$\therefore \qquad \int_0^T g(t) \sin \omega_n t\ \delta t \;=\; \sum_{n=1}^{\infty} \frac{a_n}{2} T$$

$$\therefore \qquad a_n \;=\; \frac{2}{T} \int_0^T g(t) \sin 2\pi n f t\ \delta t$$

The b_n amplitude can be computed for any given g(t) by multiplying both sides of equation (1) by cos (2πkft) and integrating with 0 to T, we get

$$\therefore \qquad \int_0^T g(t) \cos(2\pi k f t)\,\delta t \;=\; \int_0^T \frac{c}{2} \cos(2\pi k f t)\,\delta t \sum_{n=1}^{\infty} a_n \int_0^T \sin(2\pi n f t).\ \cos\ (2\pi\ kft)\ \delta t$$

$$+ \sum_{n=1}^{\infty} b_n \int_0^T \cos(2\pi n f t). \cos(2\pi k f t)\ \delta t \qquad \text{...(5)}$$

Suppose ω = 2πf and k = n, ∴ equation (5) becomes

$$\therefore \; \int_0^T g(t)\,\cos \omega_n t\ \delta t \;=\; \underset{\xrightarrow[\text{1}^{\text{st}}\text{ term}]{}}{\int_0^T \frac{c}{2} \cos \omega_n t\ \delta t} + \sum_{n=1}^{\infty} a_n \underset{\xrightarrow[\text{2}^{\text{nd}}\text{ term}]{}}{\int_0^T \sin \omega_n t. \cos \omega_n t\ \delta t} + \sum_{n=1}^{\infty} b_n \underset{\xleftrightarrow[\text{3}^{\text{rd}}\text{ term}]{}}{\int_0^T \cos^2 \omega_n t\ \delta t} \qquad \text{...(6)}$$

1st term:

$$= \int_0^T \frac{c}{2} \cos \omega_n t\ \delta t$$

$$= \frac{c}{2} \left[\frac{\sin \omega_n t}{w_n} \right]_0^T$$

$$= \frac{c}{2w_n} [\sin \omega_n t - \sin \omega_n 0] = 0$$

2nd term:

$$= \int_0^T \sum_{n=1}^{\infty} a_n \sin \omega_n t. \cos \omega_n t\ \delta t$$

$$= \sum_{n=1}^{\infty} a_n \left[\frac{\sin 2\,\omega_n t - \sin 2\,\omega_n 0}{2\omega_n} \right]_0^T = \sum_{n=1}^{\infty} \frac{a_n}{2\omega_n} [0] = 0$$

3rd term:

$$= \sum_{n=1}^{\infty} b_n \int_0^T \cos^2 \omega_n t\ \delta t$$

$$= \sum_{n=1}^{\infty} b_n \left[\frac{1 + \sin 2\omega_n}{2} \right]_0^T$$

$$= \sum_{n=1}^{\infty} \frac{b_n}{2} [(T - 0) + (\sin 2T - \sin 2 \times 0)]$$

$$= \sum_{n=1}^{\infty} \frac{b_n}{2}[T - 0]$$

$$= \sum_{n=1}^{\infty} \frac{b_n}{2} T$$

Substituting the values of these three terms in equation (6), we get,

$$\therefore \qquad \int_0^T g(t).\cos\omega_n t\, \delta t = 0 + 0 + \sum_{n=1}^{\infty} \frac{b_n}{2} T$$

$$\therefore \qquad b_n = \frac{2}{T}\int_0^T g(t).\cos(2\pi n ft)\,\delta t$$

And also by just integrating both sides of the equation (1), we can compute the value of c.

$$\therefore \qquad \int_0^T g(t)\,\delta t = \int_0^T \frac{c}{2}\,\delta t + \sum_{n=1}^{\infty} a_n \int_0^T \sin(2\pi n ft)\,\delta t + \sum_{n=1}^{\infty} b_n \int_0^T \cos(2\pi n ft)\,\delta t \qquad \text{...(7)}$$

1st term: $= \frac{c}{2}[T]_0^T$

$$= \frac{c}{2}[T - 0]$$

$$= \frac{c}{2} T$$

2nd term: $= \sum_{n=1}^{\infty} \frac{a_n}{2\pi n f}[\cos(2\pi n ft) - \cos(2\pi n f \times 0)] = 0$

3rd term: $= \sum_{n=1}^{\infty} \frac{b_n}{2\pi n f}[\sin(2\pi n ft) - \sin(2\pi n f \times 0)] = 0$

Substituting the values of these three terms in equation (7)

$$\therefore \qquad \int_0^T g(t)\,\delta t = \frac{c}{2} T + 0 + 0$$

$$\therefore \qquad c = \frac{2}{T}\int_0^T g(t)\,\delta t$$

6.2.2 Bandwidth Limited Signals

To see what all this has to do with data communication. Let us consider a specific example, the transmission of the ASCII character “b” encoded in an 8-bit byte. The bit pattern that is to be transmitted as 01100010. The left hand part of Figure 6.1 (a) shows the voltage output by the transmitting computer. The Fourier analysis of this signal yields the coefficients.

$$a_n = \frac{1}{\pi_n}\left[\cos(\pi_n/4) - \cos(3\pi_n/4) + \cos(6\pi_n/4) - \cos(7\pi_n/4)\right]$$

$$b_n = \frac{1}{\pi_n}\left[\sin(3\pi_n/4) - \sin(\pi_n/4) + \sin(7\pi_n/4) - \sin(6\pi_n/4)\right]$$

$$c = 3/8$$

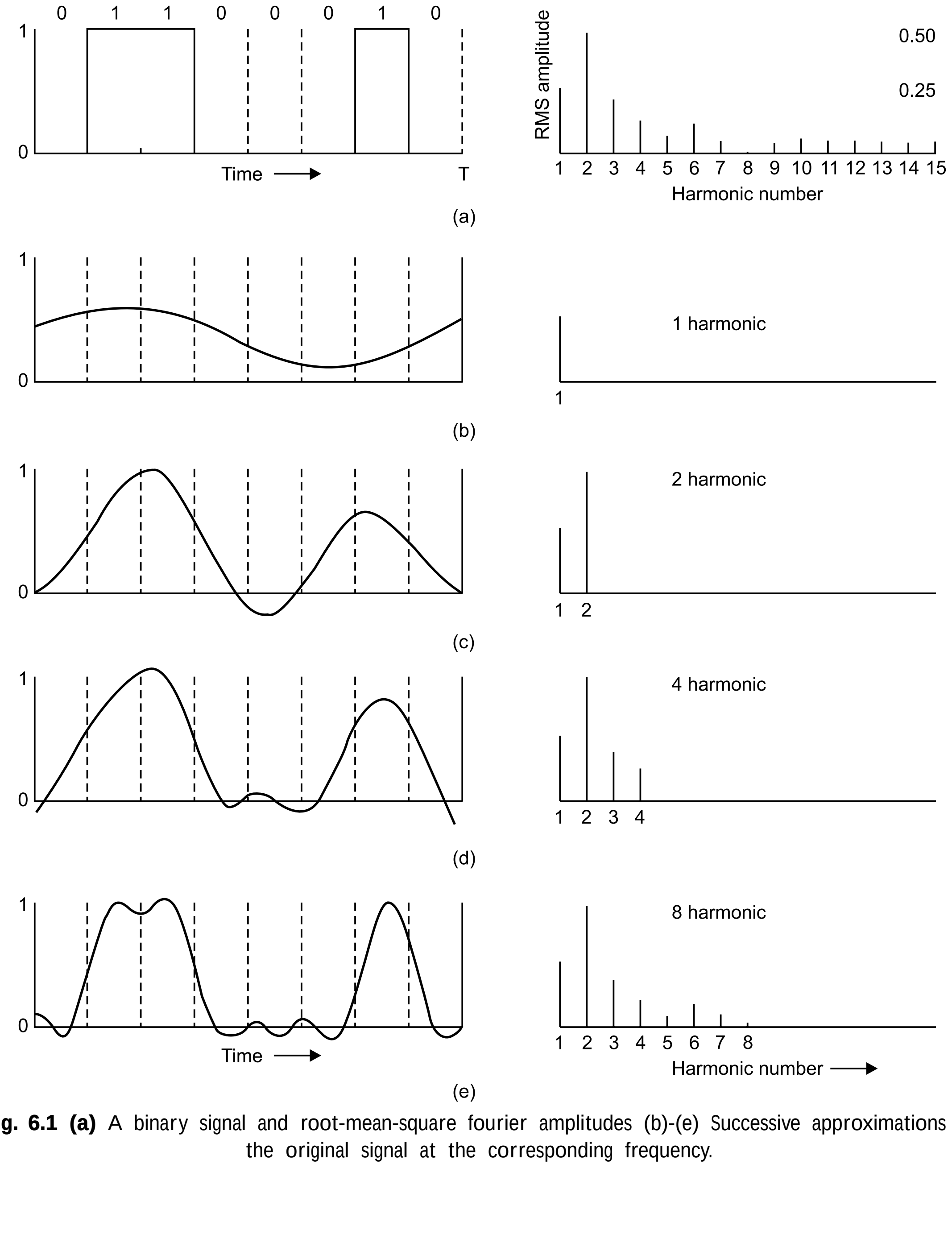


Fig. 6.1 (a) A binary signal and root-mean-square fourier amplitudes (b)-(e) Successive approximations to the original signal at the corresponding frequency.

The root mean square amplitudes, $\sqrt{a_n^2 + b_n^2}$, for the first few terms as shown on the right hand side of Figure 6.1 (a). These values are of interest because their squares are proportional to the energy transmitted.

No transmission facility can transmit signals without losing some power in the process. If all the Fourier components were equally diminished, the resulting signal would be reduced in amplitude but not distorted. (*i.e.*, it would have the same nice squared off shape as Figure 6.1 (a)). Unfortunately, all transmission facilities diminish different Fourier components by different amount, thus introducing distortion. Usually, the amplitudes are transmitted undiminished from 0 up to some Frequency Fc [measured in Cycles/sec or Hertz (Hz)] with all frequencies above this cutoff frequency strongly attenuated.

In some cases this is a physical property of the transmission medium, and in other cases a filter is intentionally introduced into the circuit to limit the amount of bandwidth available to each customer.

Now let us consider how the signal of Figure 6.1 (a) would look if the bandwidth were so low that only the lowest frequencies were transmitted (*i.e.*, the function were being approximate by the first few terms of e.g. 5). Figure 6.1 (*b*) shows the signal that results from a channel that allows only the first harmonic. (the fundamental *f*) to pass through. Similarly, Figure 6.1 (*c*)-(*e*) shows the spectre and reconstructed functions for higher bandwidth channels.

The time T required to transmit the character depends on both the encoding method and the signalling speed, the number of times per second that the signal changes its value (e.g. its voltage) the number of changes per second is measured in band. A 'b' band line does not necessarily transmit *b* bits/sec, since each signal might convey several bits. If the voltages 0, 1, 2, 3, 4, 5, 6 and 7 where used. Each signal value could be used to convey 3 bits. So the bit rate would be three times the band rate. In our example, only 0's and 1's are being used as signal levels, so the bit rate is equal to the band rate.

Given a bit rate of *b* bits/sec, the time required to send 8 bits (for example) is 8/*b* sec., so the frequency of the first harmonic is *b*/8 Hz. An ordinary telephone line, often called a voice grade line, has an artificially introduced cutoff frequency near 3000 Hz. This restriction mean that the number of the highest harmonic passed through is 3000/(*b*/8) or 24000/*b*, roughly (the cutoff is not share).

For some data rates, the numbers work out as shown in table 6.1. From these numbers, it is clear that, trying to send at 9600 bps over a voice grade telephone line will transform figure 6.1 (*a*) into something looking like figure 6.1 (c) making accurate reception of the original binary bit stream tricky. It should be obvious that at data rates much higher than 38.4 kbps there is no hope at all for binary signals, even if the transmission facility is completely noiseless. In other words limiting the bandwidth limits the data rate. Even for perfect channels. However, sophisticated coding schemes that use several voltage levels do exist and can achieve higher data rates.

Table 6.1 Relation between data rate and harmonics

BPS	*T (m.sec)*	*First harmonic (Hz)*	*# Harmonic sent*
300	26.67	37.5	80
600	13.33	75	40
1200	6.67	150	20
2400	3.33	300	10
4800	1.67	600	5
9600	0.83	1200	2
19200	0.42	2400	1
38400	0.21	4800	0

6.2.3 The maximum Date Rate of a Channel

Nyquist proved that if an arbitrary signal has been run through a low pass filter of bandwidth H, the filtered signal can be completely reconstructed by making only 2H (exact) samples per second. Sampling the line faster than 2H times per second, is pointless because the higher frequency components such that sampling could recover have already been filtered out. If the signal consists of V discrete levels, **Nyquist's** theorem states that:

$$\text{Maximum data rate} = 2H \log_2 V \text{ bits/sec}$$

For example, noiseless 3 kHz channel can not transmit binary (*i.e.*, two level) signals at a rate exceeding 6000 bps.

So far we have considered only noiseless channels. If random noise is present, the situation deteriorates rapidly. The amount of thermal noise present is measured by the ratio of the signal power to the noises power called the **signal to noise ratio.** If we denote the signal power by S and the noise power by N, the signal to noise ratio is S/N. Usually the ratio itself is not quoted, instead, the quantity 10 $\log_{10}$ S/N given. These units are called decibels (dB). AB S/N ratio of 10 is 10 dB, a ratio of 100 is 20 dB, a ratio is 1000 is 30 dB and so on. The manufacturers of stereo amplifiers often characterize the bandwidth (Frequency range) over which their product is linear by giving the 3 dB frequency on each end. These are the points at which the amplification factor has been approximately halved.

Shannon's major result is that the maximum data rate of a noisy channel whose bandwidth is H Hz, and whose signal-to-noise ratio is S/N, is given by

$$\text{Maximum No. of bits/sec} = H \log_2 (1 + S/N)$$

For example, a channel of 3000 Hz bandwidth and a signal to noise ratio of 30 dB can never transmit much more than 30,000 bps, no matter how many or few signal levels are used and no matter how often or how infrequent samples are taken. Shannon's result was derived using information theory arguments and applies to any channel subject to Gaussion (thermal) noise.

6.3 TRANSMISSION MEDIA

Various physical media can be used for the actual transmission. Each one has its own niche in terms of bandwidth, delay, cost and ease of installation and maintenance.

Transmission media are roughly divided into two broad categories: guided and unguided shown in Figure 6.2.

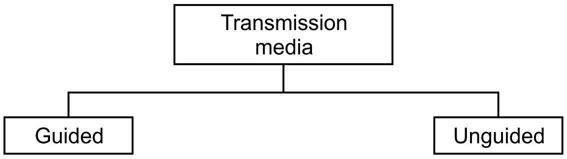


Fig. 6.2 Classes of transmission media

6.3.1 Guided Media

Guided media, which are those that provide a conduit from one device to another, include twisted-pair cable, co-axial cable, and fiber optic cable shown in Figure 6.3. A signal travelling along any of these media is directed and contained by the physical limits of the medium. Twisted-pair and co-axial cable use metallic (Copper) conductors that accept and transport signals in the form of electrical current. Optical fiber is a glass or plastic cable that accepts and transports signals in the form of light.

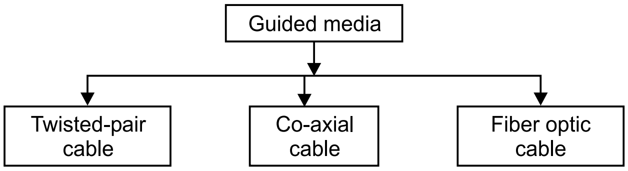


Fig. 6.3 Categories of guided media

6.3.1.1 Twisted-Pair Cable

Although the bandwidth characteristics of magnetic tape are excellent, the delay characteristics are poor. Transmission time is measured in minutes or hours, not milliseconds. For many applications an on-line connection is needed. The oldest and still most common transmission medium is twisted pair. A twisted pair consists of two insulated copper wires, typically about 1 mm thick. The wires are twisted together in a helical form just, as a DNA molecule. The twisted form is used to reduce electrical interference to similar pair close by (two parallel wires constitute a simple antenna a twisted pair does not).

The most common application of the twisted pair is the telephone system. Nearly all telephones connected to the telephone company office by a twisted pair. Twisted pairs can run several km without amplification but for longer distance, repeaters are needed when many twisted pairs run-in parallel for a substantial distance, such as all the wires coming from an apartment building to tie telephone company office, they are bundled together and encased in a protective sheath. The pairs in these bundles would interfere with one another if it were not for the twisting. In parts of the world where telephone lines run on poles above ground, it is common to see bundles several centimeters in diameter.

Twisted pairs can be used either analog or digital transmission. The bandwidth depends on the thickness of the wire and the distance travelled, but several megabits/sec can be achieved for a few km in many cases. Due to their adequate performance and low cost twisted pair wires are widely used and likely to remain so for years to come.

Twisted pair cabling comes in two varieties as **Unshielded** and **Shielded**.

Unshielded twisted-pair (UTP) cable

Unshielded twisted-pair (UTP) cable is the most common type of telecommunication medium in use today. Although most familiar from its use in telephone systems, it frequency range is suitable for transmitting both data and voice.

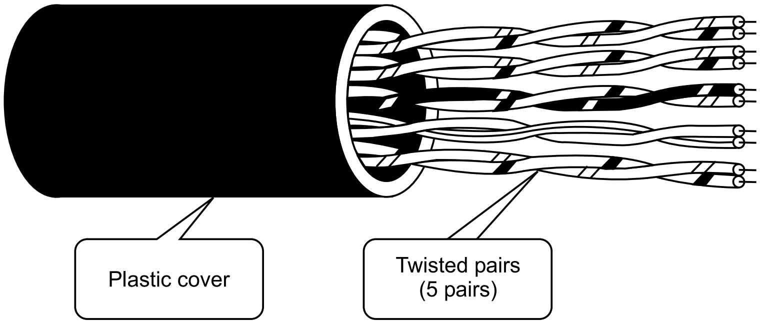


Fig. 6.4 Cable with five unshielded twisted pairs of wires

Figure 6.4 shows a cable containing five unshielded twisted pairs. The Electronic Industries Association (EIA) has developed standards to grade UTP cables by quality. Categories are determined by cable quality, with 1 as the lowest and 5 as the highest. Each EIA category is suitable for certain uses and not for others:

Category 1: The basic twisted-pair cabling used in telephone systems. This level of quality is fine for voice but inadequate for all but low-speed data communication.

Category 2: The next higher grade, suitable for voice and for data transmission of up to 4 Mbps.

Category 3: Required to have at least three twists per foot and can be used for data transmission of up to 10 Mbps. It is now the standard cable for most telephone systems.

Category 4: Must also have at least three twists per foot as well as other conditions to bring the possible transmission rate to 16 Mbps.

Category 5: Used for data transmission up to 100 Mbps.

Shielded twisted-pair (STP) cable

Shielded twisted-pair (STP) cable has a metal foil or braided-mesh covering that encases each pair of insulated conductors shown in Figure 6.5. The metal calling prevents the penetration of electromagnetic noise. It also can eliminate a phenomenon called crosstalk.

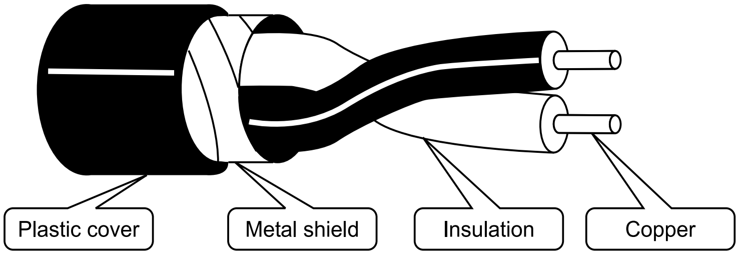


Fig. 6.5 Shielded twisted-pair cable

It occurs when one line picks up some of the signals travelling down another line. Shielding each pair of a twisted-pair cable can eliminate most crosstalk.

Advantages

1. Well understood technology
2. Easy to add computers to network
3. Least expensive medium
4. Same medium as telephone
5. Pre-existing phone wire may be in place to connect workstations.

Disadvantages

1. Susceptible to noise
2. Limited maximum bandwidth
3. Distance limitations
4. Easiest to tap
5. Requires expensive support, electronic and devices

6.3.1.2 Co-axial Cable

Another common transmission medium is the co-axial cable (known to its many friends are just "Coax"). Two kinds of co-axial cable are shown in Figure 6.6.

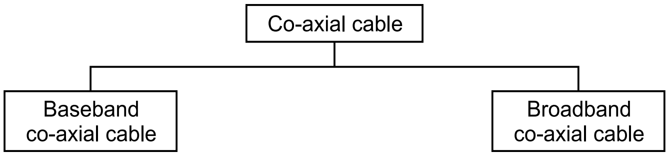


Fig. 6.6 Categories of co-axial cable

Baseband co-axial cable

This cable is about 50-ohm, used for digital transmission. A co-axial cable consists of a stiff copper wire as the core, surrounded by an insulating material. The insulator is encased by a cylindrical conductor often at a closely woven braided mesh. The outer conductor is covered in a protective plastic sheath.

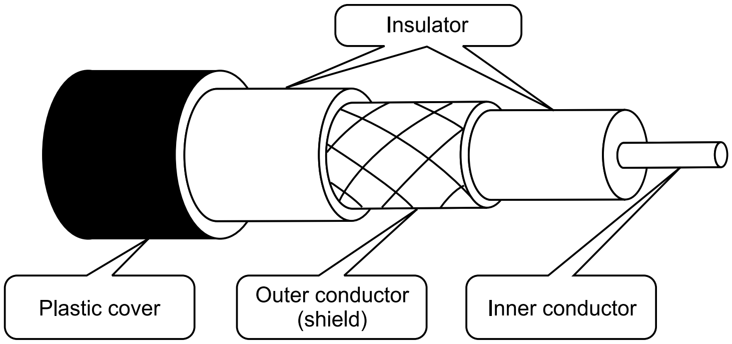


Fig. 6.7 A Cross Sectional view of co-axial cable

The construction of the co-axial cable gives it a good combination of high bandwidth and excellent noise immunity. The possible bandwidth depends on the cable length. For 1 km cables, a data rate of 10 Mbps is feasible. Higher data rates are possible on shorter cables. Longer cables can also be used but only at lower data rates. Co-axial cables are widely used for local area networks and for long distance transmission, within the telephone system.

There are **two ways** to connect computers to a co-axial cable.

- The first way is to cut the cable cleanly in two parts and insert a T junction, a connector that reconnects the cable but also provides a third wire leading off to the computer.
- The second way is to use a **Vampire tap**, which is a hole of exceedingly precise depth and width drilled into the cable, terminating in the core.

The cables used for vampire taps are thicker and more expensive than the cables used with T junctions.

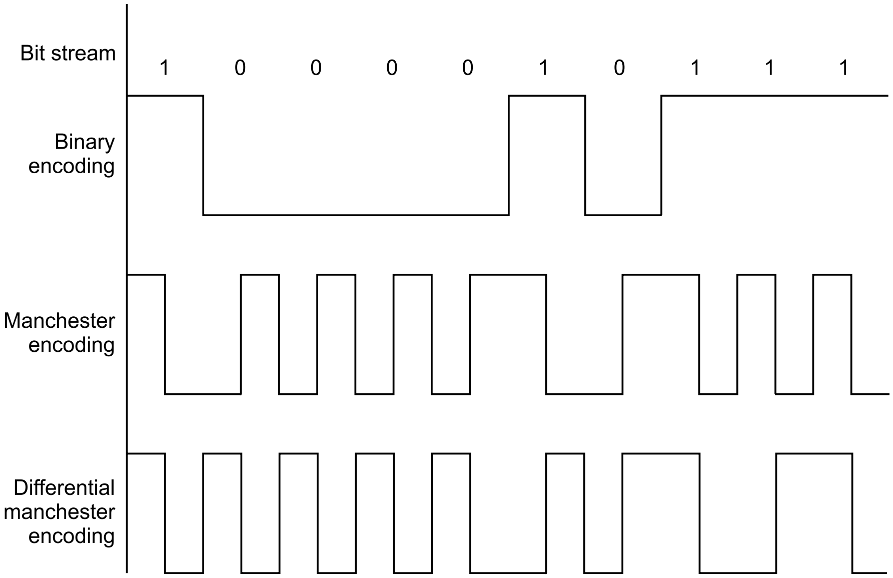


Fig. 6.8 Differential encoding schemes

Although straight binary signalling is sometimes used on co-axial cable (*e.g.* 1 volt for a 1 bit and 0 volts for a 0 bit). This method gives the receiver no way of determining when each bit starts and ends. Instead a technique called **Manchester** encoding or a related technique called **differential Manchester** encoding is preferred. With Manchester encoding, each bit period is divided into two equal intervals. A binary 1 bit is sent by having the voltage be high during the first interval and low in the second one. A binary 0 is just reverse: First low and then high. This scheme ensures that every bit period has a transition in the middle, making it easy for the receiver to synchronise with the sender. **A disadvantage** of Manchester encoding is that it requires twice as much bandwidth as straight binary encoding, because the pulses are half the width. Manchester encoding is shown in Figure 6.8.

Differential Manchester encoding is a variation of basic. Manchester encoding. In it, a 1 bit is indicated by the absence of a transition at the start of the interval. A 0 bit is indicated by the presence of a transition at the start of the interval. In both cases, there is transition in the middle as well. The differential scheme requires more complex equipment, but offers better noise immunity.

Broadband co-axial cable

This cable is about 75 ohm used for analog transmission on standard cable as television cabling.

The term “broadband” comes from the telephone world, where it refers to anything wider than 4 KHz. In the computer networking world “broadband” means any cable network using analog transmission.

Since broadband networks use standard cable television technology, the cables can be used up to 300 MHz (and sometimes up to 450 MHz) and can run for nearly 100 km due to the analog signalling, which is much less critical than digital signalling. To transmit digital signals on an analog network, each interface must contain electronics to convert the outgoing bit stream to an analog signal, and the incoming analog signal to a bit stream. Depending on the type and price of these electronics, 1 bps may occupy anywhere from 1 to 4 Hz of bandwidth. Typically a 300 MHz cable will support a total data rate of 150 Mbp.

Broadband systems are normally divided up into multiple channels, frequently the 6 MHz channels used for television broadcasting. Each channel can be used for analog television, high quality audio, or a digital bit stream at say, 3 Mbps, independent of the other channels. Television and data can be mixed on the same cable.

One key difference between baseband and broadband systems need analog amplifiers to strengthen the signal periodically. These amplifiers can only transmit signals in one direction, so a computer outputting a packet will not be able to reach computers “upstream” from it if an amplifier lies between them. To get around this problem, two types of broadband systems have been developed.

1. Dual Cable System
2. Single Cable System

1. Dual cable system

Dual cable system have two identical cables running next to each other. To transmit data, a computer outputs the data onto head, which runs to a device called the head end at the root of the cable tree. The head end then transfers the signal to cable 2 for transmissions back down the tree. All computer transmit on cable 1 and receive on cable 2. A dual cable system is shown in the Figure 6.9.

In figure, circles shows computer nodes. Amplifiers are used to amplify the signal. Dotted lines carry signals for receivers. Every computer can listen the signal but only destination can receive it.

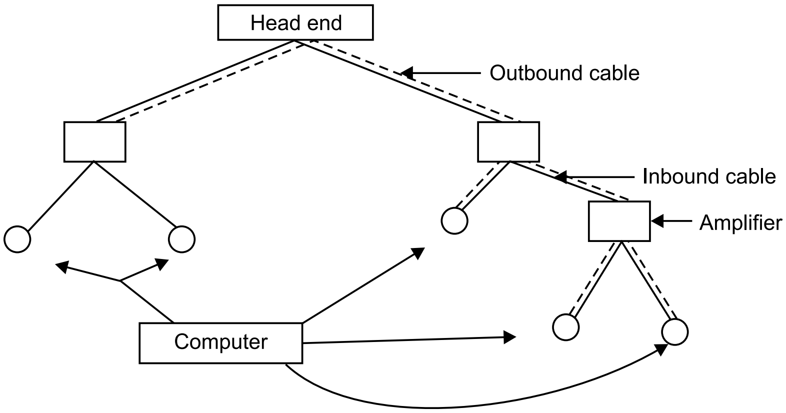


Fig. 6.9 Dual cable broadband system

2. Signal cable system

The other scheme allocates different frequency bands for inbound and outbound communication on a single cable. The low frequency band is used for communication from the computer to the head end, which then shifts the signal to the high frequency band and rebroadcasts it for destination. These techniques and frequencies were developed for cable television and have been taken over for networking as shown in Figure 6.10.

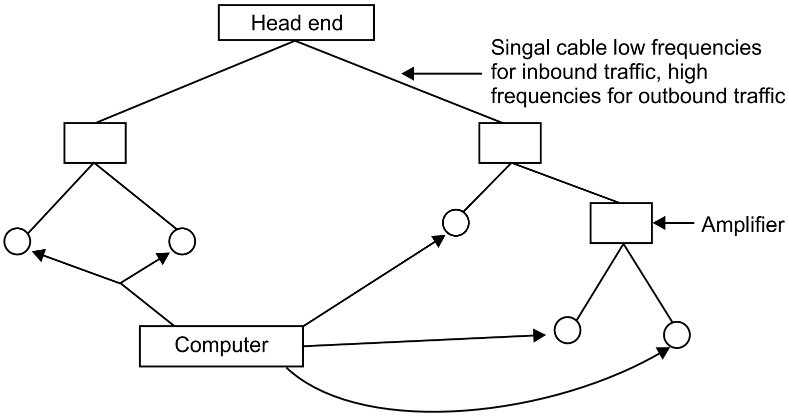

Head end
Singal cable low frequencies
for inbound traffic, high
frequencies for outbound traffic
Amplifier
Computer

Fig. 6.10 Single cable broadband system

Comparison between baseband and broadband cable

Baseband is simple and inexpensive to install and require inexpensive interfaces. It offers a single digital channel with a data rate of about 10 Mbps over a distance of 1 km. Using off-the-shelf co-axial cable. For most data communication application baseband is perfectly adequate.

Broadband, on the other hand, requires experienced radio frequency engineers to plan the cable and amplifier layout and to install the system. Skilled professionals are also required to maintain the system and periodically tune the amplifiers during its use. The head end also requires careful servicing because a failure of the head end bring the network down. The broadband interfaces are also more expensive than the baseband one.

However, broadband offers multiple channels and can transmit data, voice and television on the same cable for tens of kilometer if need be. For most applications, the additional bandwidth of broadband does not justify its complexity and expense so baseband is more widely used.

Advantages

1. Low maintenance cost.
2. Simple to install and tap.
3. Better resistance to signal noise over longer distance.

Disadvantages

1. Limited distance and topology.
2. Low security, easily tapped.
3. Difficult to make major changes to the cabling topology.
4. Cable cost is higher than that of twisted pair. Although electronic, support components are less expensive.

6.3.1.3 Optical Fiber

Recent developments in optical technology have made it possible to transmit data by pulses of light. A light pulse can be used to signal a 1 bit and the absence of a pulse signals a 0 bit. Visible light has a frequency of about 108 MHz. So the bandwidth of an optical transmission system is potentially enormous. An optical transmission systems has three components.

(*a*) ***The transmission medium:*** The transmission medium is an ultra thin fiber of glass or fused silica.

(*b*) ***The light source:*** The light source is either an LED (Light Emitting Diode) or a laser diode, both of which emit light pulses when an electrical current is applied.

(*c*) ***The detector:*** The detector is a photodiode, which generates, an electrical pulse when light falls on it.

By attaching an LED or laser diode to one end of an optical fiber and a photodiode to the other, we have a unidirectional data transmission system that accepts an electrical signal, converts and transmits it by light pulses, and then reconverts the output to an electrical signal at the receiving end.

This transmission system would leak light and be useless in practice except for an interesting principle of physics. When a light ray passes from one medium to another for example, from fused silica to air, the ray is refracted (bent) at the silicon wire boundary as shown in Figure 6.11 (a).

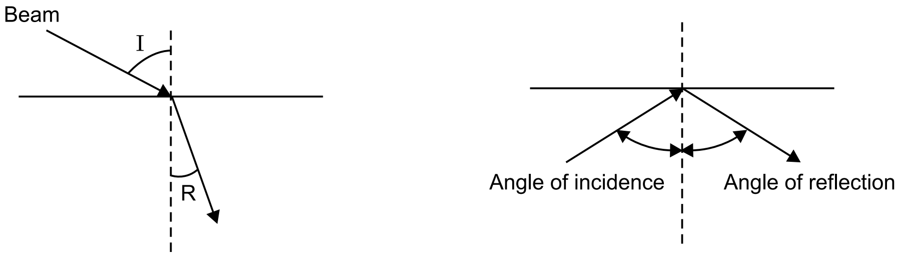


Fig. 6.11 (a) Refraction

Fig. 6.11 (b) Reflection

Here we see a light ray incident on the boundary at an angle α emerging at an angle β.

The amount of refraction depends on the properties of the two media. For angles of incidence above a certain critical value, the light is refracted back into the silica, none of it escapes into the air. Thus a light ray incident at or above the critical angle is trapped inside the fiber, as shown in Figure 6.11 (b) and can propagate for many kilometers with virtually no less.

Propagation modes

There are two modes for propagating light along optical channels, each requires fiber with different physical characteristics : Multimode and single mode shown in Figure 6.12.

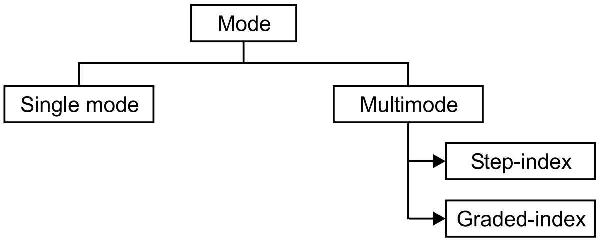


Fig. 6.12 Propagation modes

Single mode

Single mode uses step-index fiber and a highly focused source of light that limits beams to a small range of angles, all close to the horizontal. The single-mode fiber itself is manufactured with a much smaller diameter than that of multimode fibers, and with substantially lower density (index of refraction). The decrease in density results in a critical angle that is close enough to go degrees to make the propagation of beams almost horizontal.

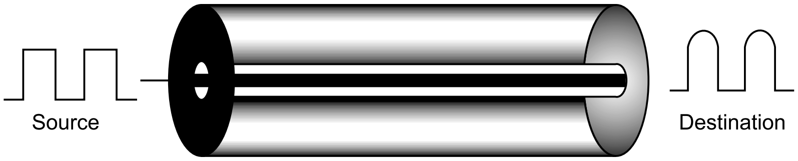


Fig. 6.13 Single-mode fiber

In this case, propagation of different beams is almost identical and delays are negligible. All of the beams arrive at the destination “together” and can be recombined without distortion to the signal. The single-mode fiber in shown in Figure 6.13.

Multimode fiber

In multimode fiber, multiple beams from a light source move through the core in different paths. How these beams move within the cable depends on the structure of the core. Multimode fibers are of two types as step-index and graded-index fiber.

Multimode step-index fiber

In multimode step-index fiber, the density of the core remains constant from the center to the edges. A beam of light moves through ‘this constant density in a straight line until it reaches the interface of the core and the cladding. At the interface, there is an abrupt change to a lower density that alter the angle of the beam’s motion. The term step-index refers to the suddenness of this change. The Figure 6.14 shows the multimode step-index fiber.

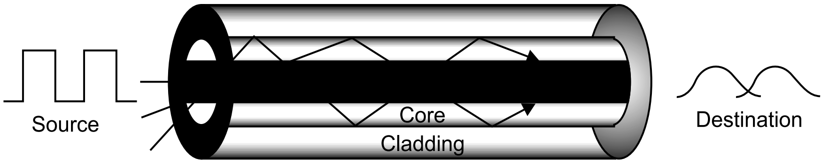


Fig. 6.14 Multimode step-index fiber

Multimode graded-index fiber

Multimode graded-index fiber, decreases distortion of the signal through the cable. The word index here refers to the index of refraction. As we know in multimode step-index fiber index of refraction is related to density. A graded-index fiber, therefore is one with varying densities. Density is highest at the center of the core and decreases gradually to its lowest at the edge. The Figure 6.15 shows the multimode graded-index fiber.

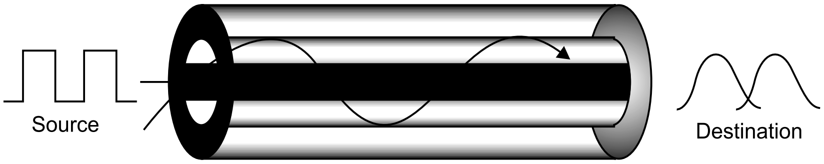


Fig. 6.15 Multimode graded-index fiber

Comparison between co-axial cable and fiber optic cable

It is instructive to compare co-axial cable to fiber optics. Fiber provides extremely high bandwidth with little power loss, so it can run for long distances between repeaters. The fibers themselves are not affected by power line surges, electromagnetic interferences, or corrosive chemicals in the air, so they can be used factory environments unsuitable for co-axial cable. Fibers are also very thin, a big plus for companies with thousands of cables and cable ducts.

On the minus side, fiber optics is an unfamiliar technology requiring skills most network engineers do not have. Fibers are difficult to splice and even more difficult to tap. Fiber networks are also inherently unidirectional, and the interfaces are considerably more expensive than electrical interfaces. However, the advantage of fiber optics are so great that much work is being directed to improving the technology and reducing the cost.

Advantages

1. High bit rates.
2. Lowest transmission loss over longer distance.
3. No subject to interference.
4. Good for network “backbones” point-to-point.
5. Supports voice, data, video.
6. Difficult to tap.

Disadvantages

1. More expensive than other cabling.
2. Cabling is inflexible; can’t be bent sharp.
3. Lack of standard components.
4. Limited (Practically) to high traffic.
5. Requires skilled installation and maintenance.
6. High installation costs.

6.3.2 Unguided Media

Unguided media, or wireless communication, transport electromagnetic waves without using a physical conductor. Instead, signals are broadcast through air (or, in a few cases, water), and thus are available to anyone who has a device capable of receiving them.

6.3.2.1 Radio Frequency Allocation

The part of electromagnetic spectrum referred as radio communication is divided into eight ranges, called bands, each regulated by government authorities. These bands are rated from very low frequency (VLF) to extremely high frequency (EHF). Figure 6.16 shows all bands and their frequency.

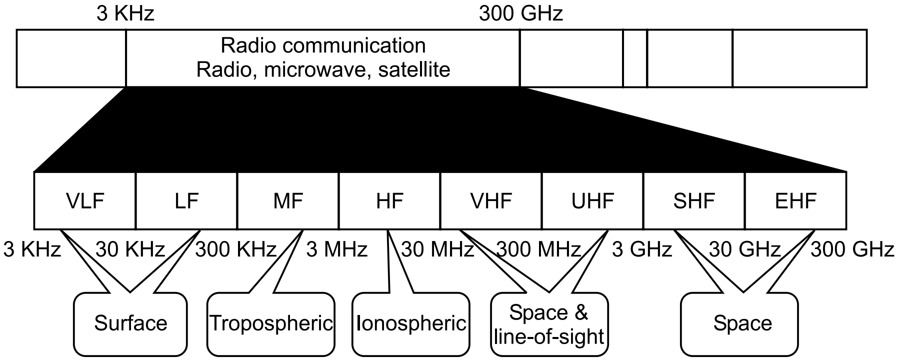


Fig. 6.16 Radio communication band

Propagation of radio waves

Radio waves transmission utilizes five different types of propagation : surface, tropospheric, ionospheric, line-of-sight, and space shown in Figure 6.17.

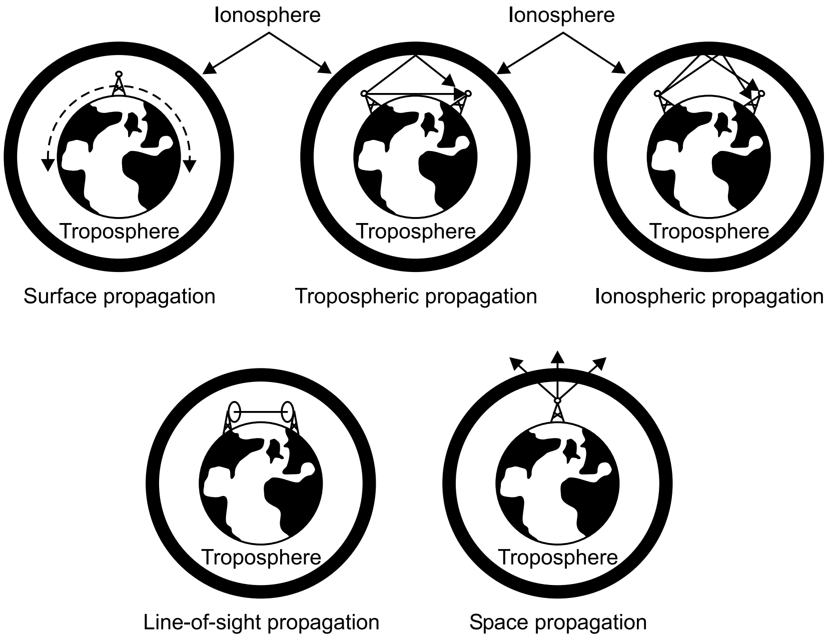


Fig. 6.17 Types of propagation

Radio technology considers that the earth is surrounded by two layers of atmosphere: the troposphere and the ionosphere.

Troposphere: The troposphere is the portion of the atmosphere extending outward approximately 30 miles from the earth's surface (in radio terminology, the troposphere includes the high-altitude layer called the stratosphere) and contains what we generally think of as air, clouds, wind, temperature variations and weather in general occur in the troposphere, as does jet plane travel.

Ionosphere: The ionosphere is the layer of atmosphere above the troposphere but below space. It is beyond what we think of as atmosphere and contains free electrically charged particles.

Surface Propagation: In surface propagation, radio waves travel through the lowest portion of the atmosphere, hugging the earth. At the lowest frequencies, signals emanate in all directions from the transmitting antenna and follow the curvature of the planet. Surface propagation can also take place in seawater.

Tropospheric Propagation: Tropospheric propagation can work in two ways. Either a signal can be directed in a straight line from antenna (line-of-sight), or it can be broadcast at an angle into the upper layers of the troposphere where it is reflected back down to the earth's surface.

Line-of-Sight Propagation: In line-of-sight propagation, very high frequency signals are transmitted in straight lines directly from antenna to antenna. Line-of-sight propagation is tricky because radio transmissions cannot be completely focused. Waves emanate upward and downward as well as forward and can reflect off the surface of the earth or parts of the atmosphere. Reflected waves that arrive at the receiving antenna later than the direct portion of the transmission can corrupt the received signal.

Ionospheric Propagation: In ionospheric propagation, higher-frequency radio waves radiate upward into the ionosphere where they are reflected back to earth. This type of transmission allows for greater distances to be covered with lower power output.

Space Propagation: Space propagation utilizes satellite relays in place of atmospheric refraction. A broadcast signal is received by an orbiting satellite, which rebroadcasts the signal to the intended receiver back on the earth. Satellite transmission is basically line-of-sight with an intermediary.

Propagation of Specific Signals: The type of propagation used in radio trans-depends on the frequency (speed) of the signal. Each frequency is suited for a specific layer of the atmosphere and is most efficiently transmitted and received by technologies adapted to that layer.

6.3.2.2 Terrestrial Microwave

Microwaves do not follow the curvature of the earth and therefore require line-of-sight transmission and reception equipment. The distance coverable by a line-of-sight signal depends to a large extent on the height of the antenna: the taller the antennas, the longer the sight distance.

Microwave signals propagate in one direction at a time, which means that two frequencies are necessary for two-way communication such as a telephone conversation. One frequency is reserved for microwave transmission in one direction and the other for transmission in the other. Each frequency requires its own transmitter and receiver. Today, both pieces of equipment usually are combined in a single piece of equipment called a trans receiver, which allows a single antenna to serve both frequencies and functions.

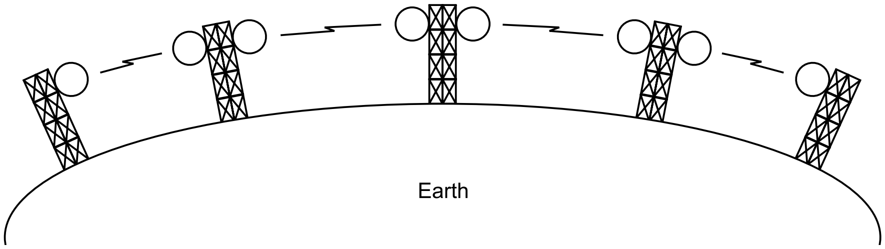


Fig. 6.18 Terrestrial microwave

Repeaters

To increase the distance served by terrestrial microwave, a system of repeaters.

Outgoing transmissions are broadcast through a horn aimed at the dish. The microwaves hit the dish and are deflected outward in a reversal of the receipt path.

Horn antenna

A horn antenna looks like a gigantic scoop. Outgoing transmissions are broadcast up a stem (resembling a handle) and deflected outward in a series of narrow parallel beams by the curved head as shown in Figure 6.20. Received transmissions are collected by the scooped shape of the horn, in a manner similar to the parabolic dish, and are deflected down into the stem.

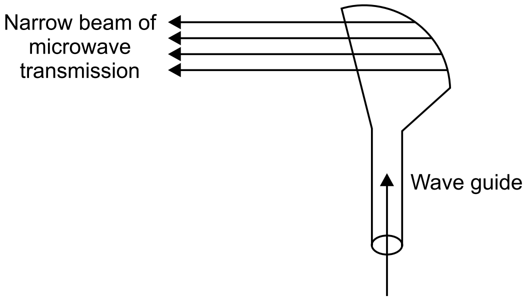


Fig. 6.20 Horn antenna

6.3.2.3 Satellite Communication

Satellite transmission is much like line-of-sight microwave transmission in which one of the stations is a satellite orbiting the earth. The principle is the same as terrestrial microwave, with a satellite acting as a supertall antenna and repeater as shown in Figure 6.21. Although in satellite transmission signals must still travel in straight lines, the limitations imposed on distance by the curvature of the earth are reduced. In this way, satellite relays allow microwave signals to span continents and oceans with a single bounce.

Satellite microwave can provide transmission capability to and from any location on earth, no matter how remote. This advantage makes high-quality communication available to undeveloped parts of the world without requiring a huge investment in ground-based infrastructure. Satellites themselves are extremely expensive, of course, but leasing time or frequencies on one can be relatively cheap.

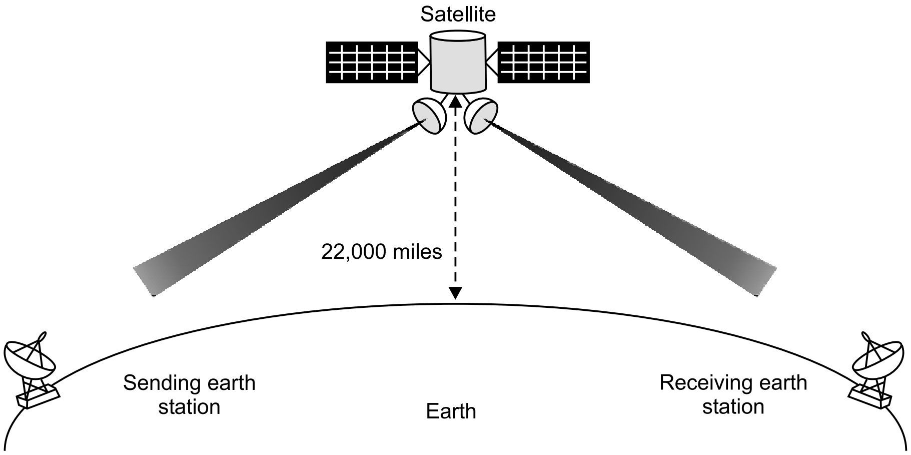


Fig. 6.21 Satellite communication

An artificial satellite needs to have an **orbit** the path in which it travels around the earth. The orbit can be equatorial, inclined or polar. The time required for a satellite to make a complete trip around the earth, is referred as the period of a satellite, which defines the period as a function of the distance of the satellite from the center of the earth.

$$\text{Period} = c * \text{distance}^{1.5}$$

where,

c = constant $\left(\cong 1/100\right)$

period is in second,

distance is in kilometers

Categories of satellites

Based on the location of the orbit, satellites can be divided into three categories: GEO, LEO, and MEO as shown in Figure 6.22.

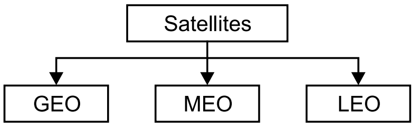


Fig. 6.22 Categories of satellite

Frequency Bands for Satellite Communication

The frequencies reserved for satellite microwave communication are in the gigahertz (GHz) range. Each satellite sends and receives over two different bands. Transmission from the earth to the satellite is called **uplink**. Transmission from the satellite to the earth is called **downlink.** Table 6.2 shows the band names and frequencies for each range.

Table 6.2 Satellite frequency bands

Band	*Downlink, GHz*	*Uplink, GHz*	*Bandwidth, MHz*
L	1.5	1.6	15
S	1.9	2.2	70
C	4	6	500
Ku	11	14	500
Ka	20	30	3500

GEO Satellites: Line-of-sight propagation requires that the sending and receiving antennas be locked onto each other's location at all times. To ensure constant communication, the satellite must move at the same speed as the earth so that it seems to remain fixed above a certain spot. Such satellites are called as **geosynchronous**.

One geosynchronous satellite cannot cover the whole earth. One satellite in orbit has line-of-sight contact with a vast number of stations, but the curvature of the earth still keeps much of the planet out of sight. It takes a minimum of three satellites equidistant from each other in **geosynchronous** Earth orbit (GEO) to provide full global transmission Figure 6.23 shows three satellites, each 120° from another in geosynchronous orbit around equator.

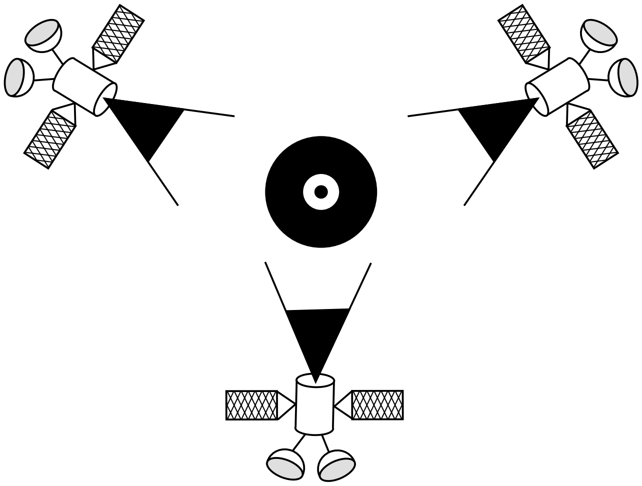


Fig. 6.23 Satellites in geosynchronous orbit

MEO Satellite: Medium-Earth Orbit (MEO) satellites are positioned between the two Van Allen belts. A satellite at this orbit takes approximately 6 hours to circle the earth.

One example of a MEO satellite system is the Global Positioning System (GPS) orbiting at an altitude about 18,000 km (11,000 miles) above the earth. The GPS system consists of 24 satellites and is used for land and sea navigation to provide time and locations for vehicles and ships. GPS is based on the principle called **triangulation**.

GPS uses 24 satellites in six orbits, as shown in Figure 6.24. The orbits and the locations of the satellites in each orbit are designed in such a way that, at any time. Four satellites are visible from any point on Earth. A GPS receiver has an almanac that tells the current position of a satellite. It then sends a signal to four satellites and measures how long it takes for the signal to return. It calculates your position on the Earth. A GPS receiver can also show you where you are on a map.

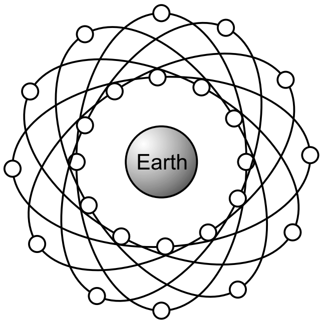


Fig. 6.24 GPS

LEO Satellites: Low-Earth Orbit (LEO) satellites have polar orbits. The altitude is between 500 to 2000 km, with a rotation period to 120 min. The satellite has a speed of 20,000 to 25,000 km/h. LEO system usually has a cellular type of access.

A LEO system is made of a constellation of satellites that work together as a network, each satellite acts as a switch. Satellites that are close to each other are connected through intersatellite links (ISLs). A mobile system communicates with the satellite through a user mobile link (UML). A satellite can also communicate with an earth station through a gateway link (GWL). Figure 6.25 shows a typical LEO satellite network.

The little LEOs are mostly used for low-data-rate messaging. Global star and Iridium systems are examples of big LEOs. Teledesic is an example of broadband LEO system.

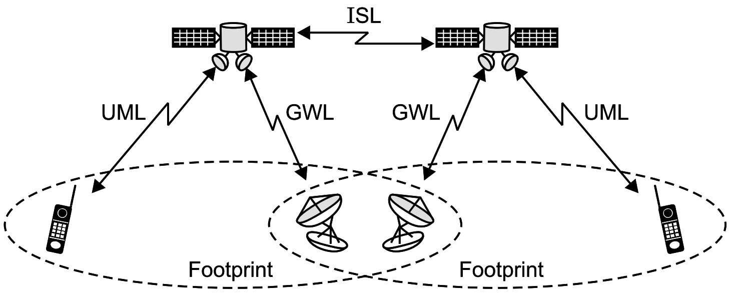


Fig. 6.25 Satellite system

6.3.2.4 Cellular Telephone

Cellular telephony provides communications between two moving units, called mobile stations (MSs), or between one mobile unit and one stationary unit, often called a land unit. A service provider must be able to locate and track a caller, assign a channel to the call, and transfer the channel from base station to base station as the caller moves out of range.

For this, each cellular service area is divided into small regions called cells. Each cell contains an antenna and is controlled by a small office, called the base station (BS). Each base station is controlled by a switching office, called a mobile switching center (MSC). The MSC coordinates communication between all the base stations and the telephone central office. It is a computerized center that is responsible for connecting calls, recording call information, and billing as shown in Figure 6.26.

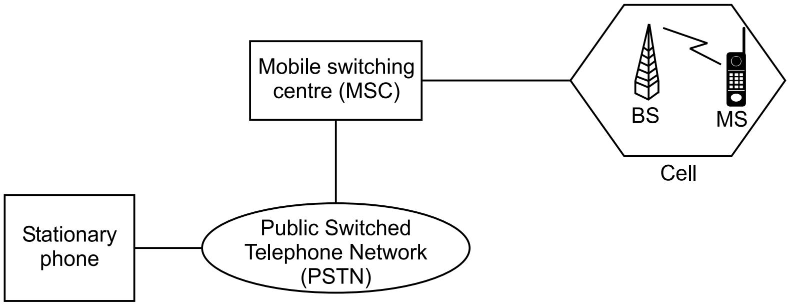

Mobile switching centre (MSC)
BS
MS
Cell
Stationary phone
Public Switched Telephone Network (PSTN)

Fig. 6.26 Cellular system

Frequency-reuse principle

Basically, neighbouring cells cannot use the same set of frequencies for communication because it may create interference for the users located near the cell boundaries. However, the set of frequencies available is limited, and frequencies need to be reuses. A frequency reuse pattern is a configuration of N cells N being the **reuse factor**, in which each cell uses a unique set of frequencies. When the pattern is repeated, the frequencies can be reused. There are several different patterns as shown in Figures 6.27 (a) and (b).

Frequency reuse pattern

Reuse factor of 4:

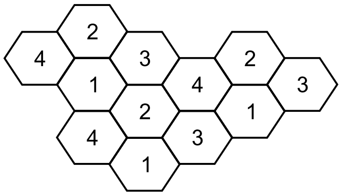


Fig. 6.27 (a)

In the above pattern, only one cell separates the cells using the same set of frequencies.

Reuse factor of 7:

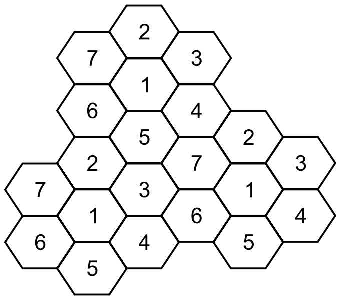


Fig. 6.27 (b) Frequency reuse patterns

In this pattern, two cells separate the reusing cells.

The cells with the same number in a pattern can use the same set of frequencies called as **reusing cells**.

Transmitting

To place a call from a mobile station, the caller enters a code of 7 or 10 digits and presses the send button. The mobile station then scans the band, seeking a setup channel with a strong signal, and sends the data to the closest base station using that channel. The base station relays the data to the MSC. The MSC sends the data on to the telephone centre office. If the called party is available, a connection is made and the result is relayed back to the MSC. The MSC assigns an unused voice channel to the call, and a connection is established.

Receiving

When a mobile phone is called, the telephone central office sends the number to the MSC. The MSC searches for the location of the mobile station by sending query signals to each cell in a process called **paging.** Once the mobile station is found, the MSC transmits a ringing signal and, when the mobile station answers, assigns a voice channel to the call, allowing voice communication to begin.

Handoff

It may happen, during a conversation, the mobile station moves from one cell to another. When it does, the signal may become weak. To solve this problem, the MSC monitors the level of the signal every few seconds. If the strength of the signal diminishes, the MSC seeks a new cell that can better accommodate the communication. The MSC then changes the channel carrying the call.

Hard Handoff

Early systems used a hard handoff. In this, a mobile station only communicates with one base station. When the MS moves from one cell to another, communication must first be broken with previous base station before communication can be established with the new one. This may create a rough transition.

Soft Handoff

New systems use a soft handoff. In this, a mobile station can communicate with two base stations at the same time.

Roaming

One feature of cellular telephony is called roaming. Roaming means a user can have access to communication or can be reached where there is coverage. A service provider usually has limited coverage. Neighbouring service providers can provide extended coverage through roaming contract.

Cellular Telephony Generation

The information about the cellular telephony generations must be shown in the table 6.3.

Table 6.3 Cellular telephony generation

Generation	*Design*	*Example*
Ist	Voice communication using analog signals	AMPS
IInd	Digitized voice	D-AMPS, GSM, CDMA
IIIrd	Digital data and voice communication	IMT-2000

6.4 TRANSMISSION IMPAIRMENT

Transmission media are not perfect. The imperfections cause impairment in the signal sent through the medium. This means that the signal at the beginning and end of the medium are not the same transmitted data is not identical to received data. Three types of impairment usually occur: attenuation, distortion and noise as shown in Figure 6.28.

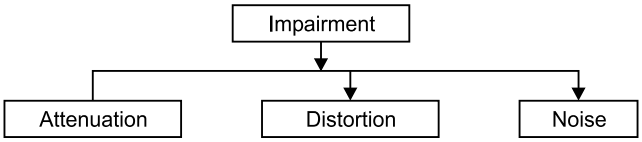


Fig. 6.28 Impairment types

Attenuation

Attenuation means loss of energy. When a signal, simple or complex, travels through a medium, it loses some of its energy so that it can overcome the resistance of the medium. That is why a wire carrying electrical signals gets warm, if not hot, after a while. Some of the electrical energy in the signal is converted to heat. To compensate for this loss, amplifiers are used to amplify the signal. Fig 6.29 shows the effect of attenuation and amplification.

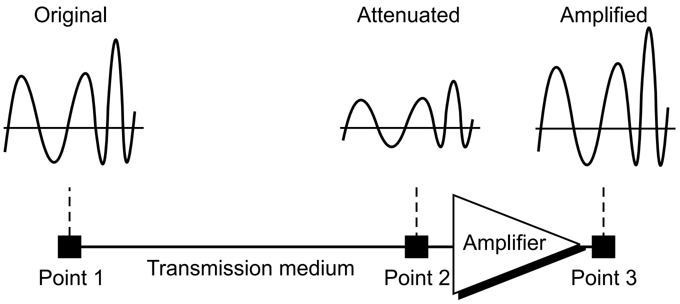


Fig. 6.29 Attenuation

Distortion

Distortion means that the signal changes its form or shape. Distortion occurs in a composite signal, made of different frequencies. Each signal component has its own propagation speed through a medium and, therefore, its own delay in arriving at the final destination. Figure 6.30 shows the effect of distortion on a composite signal.

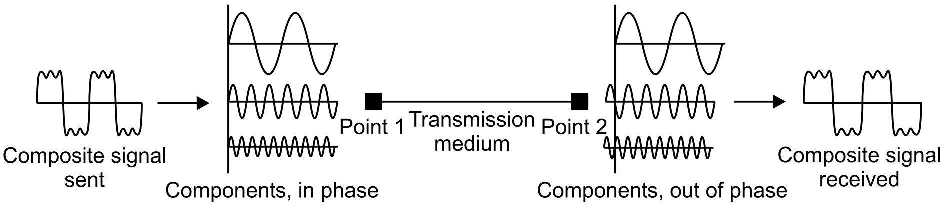


Fig. 6.30 Distortion

Noise

Noise is another problem, which is unwanted signal. Several types of noise such as thermal noise, crosstalk, impulse noise may corrupt the signal. Thermal noise is the random motion of electrons in a wire that creates an extra signal not originally sent by the transmitter. Crosstalk is the effect of one wire on the other. One wire acts as a sending antenna and the other acts as the receiving antenna. Impulse noise is a spike that comes from power lines, lightning, and so on. Figure 6.31 shows the effect of noise on a signal.

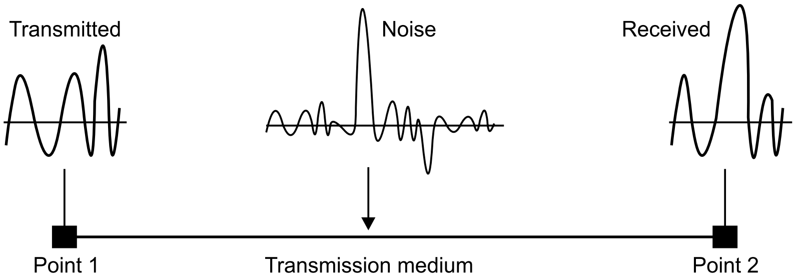


Fig. 6.31 Noise

6.5 DESIGN ISSUES OF PHYSICAL LAYER

BASIC REQUIREMENT FOR EXCHANGING BITS

The basic requirements for the devices to be able to exchange bits.

Let us consider a simple data communication situation shown in Figure 6.32, where two digital devices A and B need to example data bits.

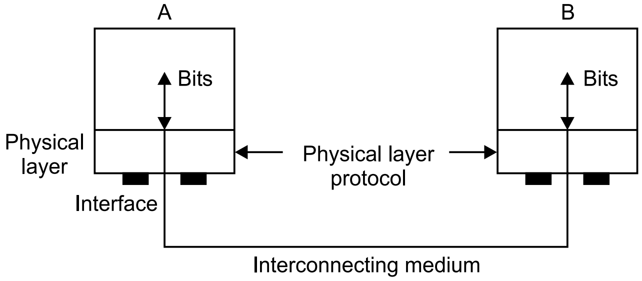


Fig. 6.32 Transmission of bits by the physical layer

1. There should be a physical interconnecting medium which can carry electrical signals between the two devices.
2. The bits need to be converted into electrical signals and vice versa.
3. The electrical signal should have characteristics (voltage, current, impedance, rise time, etc.) suitable for transmission over the medium.
4. The devices should be prepared to exchange the electrical signals.

These requirements, which are related purely to the physical aspects of transmission of bits, are met out by the physical layer.

Physical Connection

The physical layer receives the bits to be transmitted from the Data Link layer shown in Figure 6.34. At the receiving end, the physical layer hands over these bits to the Data Link layer. Thus, the physical layers at the two ends provide a transport service from one Data Link layer to the other over a "physical connection" activated by them. A physical connection is different from a physical transmission path in the sense that it is at the bit level while the transmission path is at the electrical signal level.

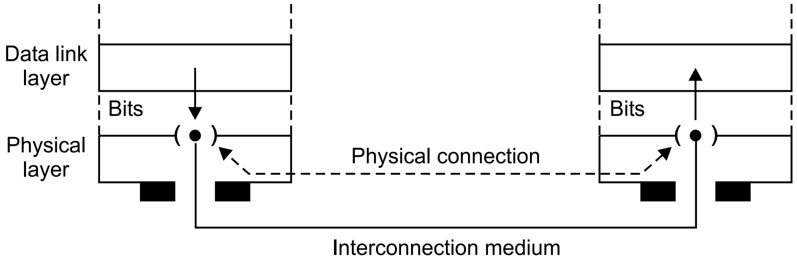


Fig. 6.33 Physical Connection

The physical connection shown in Figure 6.33 is point-to-point. Point-to-multipoint physical connection is also possible as shown in Figure 6.34.

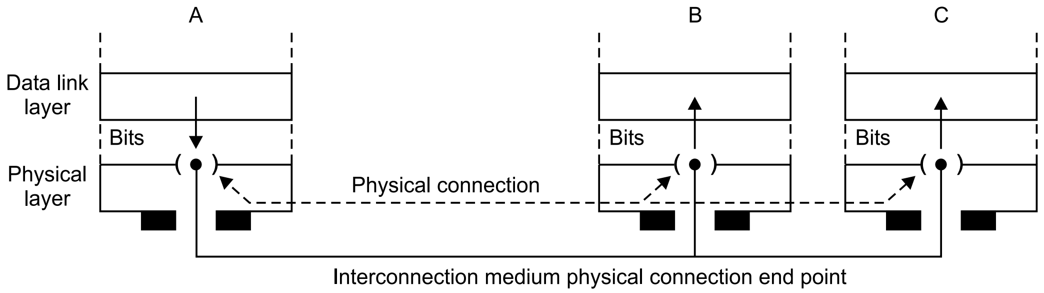


Fig. 6.34 Point-to-Multipoint physical connection

Basic Service Provided to the Data Link Layer

The basic service provided by the physical layer to the Data Link layer is the bits transmission service over the physical connection, some of the feature are as follows:

Activation/Deactivation of the Physical Connection

The physical layer, when requested by the Data Link layer, activates and deactivates a physical connection for transmission of bits. Activation ensures that if one user initiates transmission of bits, the receiver at the other end ready to receive them. The activation and deactivation service is non-confirmed, *i.e.*, the user activating or deactivating a connection is not given any feedback of the action having been carried out by the physical layer.

Transparency

The physical layer provides transparent transmission of the bit stream between the Data Link entities over the physical connection. Transparency implies that any bit sequence can be transmitted without any restriction imposed by the physical layer.

Physical Service Data Units (Ph-SDU)

Ph-SDU received from the Data Link layer consists of one bit in serial transmission and of “n” bits in parallel transmission.

Sequenced Delivery

The physical layer tries to deliver the bits in the same sequence as they were received from the Data Link layer but it does not carry out any error control. Therefore, it is likely that some of the bits are altered, some are not delivered at all, and some are duplicated.

Fault Condition Notification

Data Link entities are notified in case of any fault detected in the physical connection.

Service Primitives

The physical layer provides a non-confirmed service to the Data Link layer. The service names and primitives for activation of the physical connection data transfer and for deactivation of the physical connection are shown in Figure 6.35.

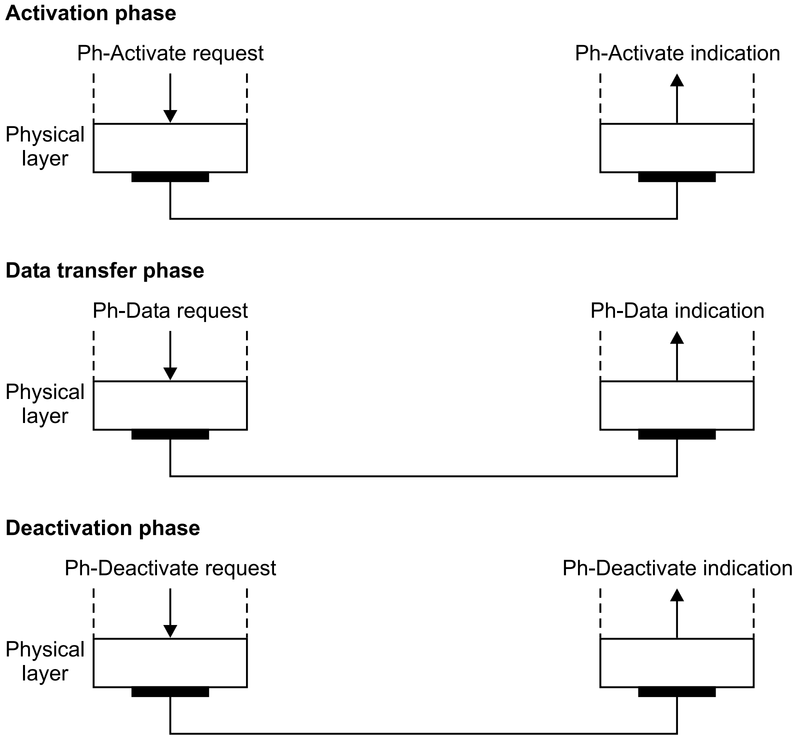


Fig. 6.35 Service primitives of the physical layer

Physical Layer Standards

Physical layer specification can be of following types as shown in Figure 6.36. These are

1. Mechanical Specification
2. Electrical Specification
3. Functional Specification
4. Procedural Specification

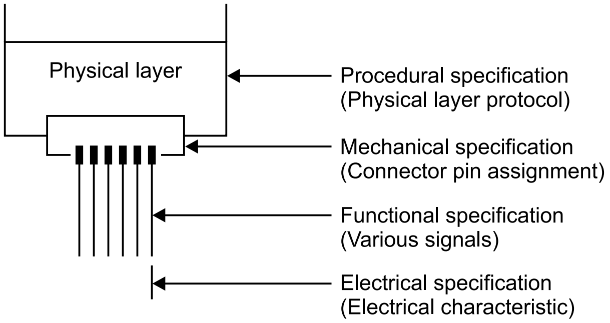

Physical layer
Procedural specification (Physical layer protocol)
Mechanical specification (Connector pin assignment)
Functional specification (Various signals)
Electrical specification (Electrical characteristic)

Fig. 6.36 Physical layer specifications

Mechanical Specification

It gives details of the mechanical dimensions and the type of connectors to be used on the device and the medium pin assignments of the connector are also specified.

Electrical Specification

Define the permissible limits of the electrical signals appearing at the interface in terms of voltages, currents, impedances, rise time, etc. The required electrical characteristics of the medium are also specified.

Functional Specification

Indicates the sequence of various control signals.

Procedural Specification

Indicates the sequence in which the control signals are exchanged between the physical layers for carrying out their functions.

Some examples of physical layer standards are as

EIA: EIA-232-D; RS-449; RS-422-A, RS-423-A

CCITT: X.20; X.20-bis; X.21; X.21-bis; V.35; V.24; V.28

ISO: ISO 2110

6.6 EIA-232-D DIGITAL INTERFACE

The EIA-232-D digital interface of Electronics Industries Association (EIA) is the most widely used physical medium interface. RS-232-C is the older and more familiar version of EIA-232-D. EIA-232-D is applicable to the following modes of transmission.

- Serial transmission of data
- Synchronous and asynchronous transmission
- Point-to-point and point-to-multipoint working
- Half duplex and full duplex transmission.

DTE/DCE Interface

EIA-232-D is applicable to the interface between a Data Terminal Equipment (DTE) and a Data Circuit Terminating Equipment (DCE) as shown in Figure 6.37. The terminal devices are usually called Data Terminal Equipment (DTE). The DTEs are interconnected using two intermediary devices which carry out the relay function. The intermediary devices are categorized as Data Circuit Terminating Equipment (DCE). They are so called because standing at the physical layer of a DTE and facing the data circuit. One finds oneself looking at an intermediary device which terminates the data circuit.

Two types of physical layer interfaces are involved.

1. Interface between a DTE and a DCE
2. Interface between the DCEs.

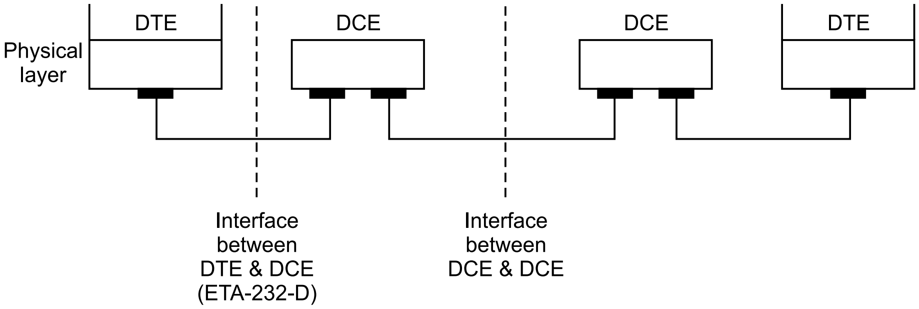

DTE
DCE
DCE
DTE
Physical layer
Interface between DTE & DCE (ETA-232-D)
Interface between DCE & DCE

Fig. 6.37 DTE/DCE Interfaces at the physical layer

DTE and DCE Ports

Over the years, use of the terms DTE and DCE for classifying two kinds of devices on the basis of their functions.

The use of the terms DTE and DCE these days at the physical layer level refers to the description of the transmission part of a device rather than the device itself.

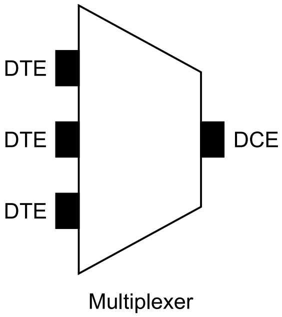


Fig. 6.38 Multiplexer with DTE and DCE ports

EIA-232-D, however, was designed with modem as DCE, and the terminology which has been used to specify its signals and functions also refers to modem as DCE.

DCE-DCE Connection

A DCE has two interfaces, DTE-side interface which is EIA-232-D, and the line-side interface which interconnects the two DCEs through the transmission medium. Several forms of connection and modes of transmission between the DCEs is shown in Figure 6.39.

1. The two DCEs may be connected directly through a dedicated transmission medium.
2. The two DCEs may be connected to PSTN.
3. The connection may be on a 2-wire transmission circuit or on a 4-wire transmission circuit.

4. The mode of transmission between the DCEs may be either full duplex or half duplex.

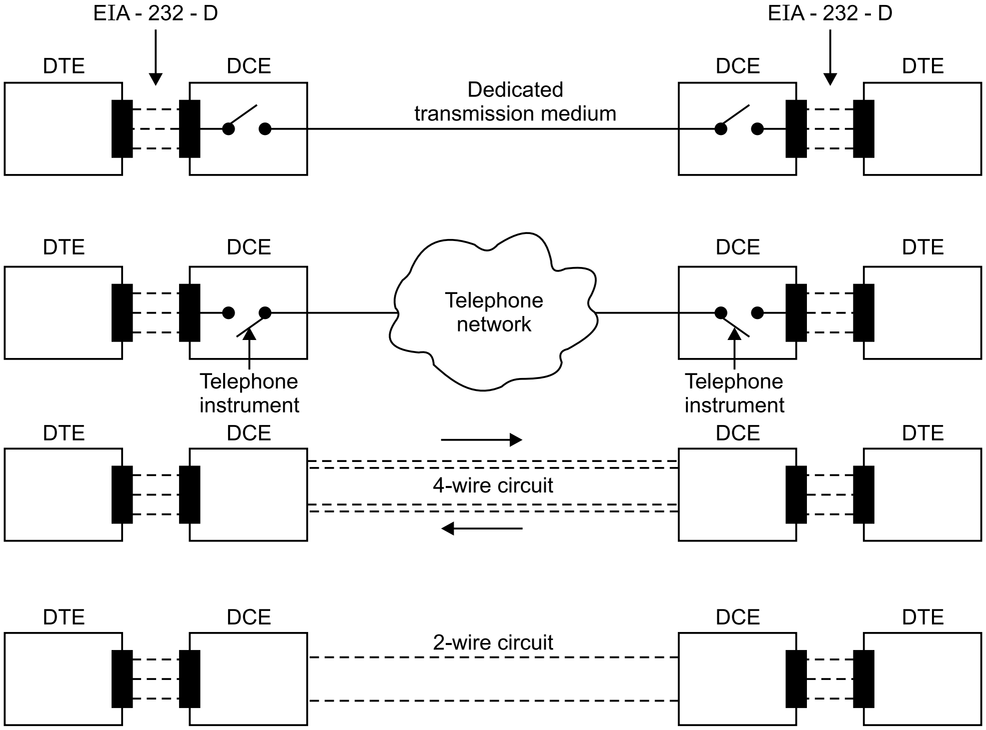


Fig. 6.39 Transmission alternatives between two DCEs

Full duplex mode of transmission is easily implemented on a 4-wire circuit. Two wires are used for transmission in one direction and the other two in the opposite direction. Full duplex operation on a 2-wire circuit requires two communication channels which are provided at different frequencies on the same medium.

6.7 EIA-232-D INTERFACE SPECIFICATIONS

EIA-232-D interface has four sets of specification for the interface between a DTE and a DCE:

* Mechanical Specifications	as per ISO 2110
* Electrical Specifications	V.28
* Functional Specifications	V.24
* Procedural Specifications	V.24

As per ISO 2110, V.28, V.24, V.24 are CCITT recommendations for the physical interface respectively. These recommendations are equivalent to EIA-232-D.

Mechanical Specifications

Mechanical specifications include mechanical design of the connectors which are used on the equipment and the interconnecting cables, and pin assignments of the connectors.

EIA-232-D defines the pin assignments and the connector design is as per ISO 2110 standard. A DB-25 connector having 25 pins as shown in Figure 6.40. The male connector is used for the DTE port and the female connector is used for the DCE port.

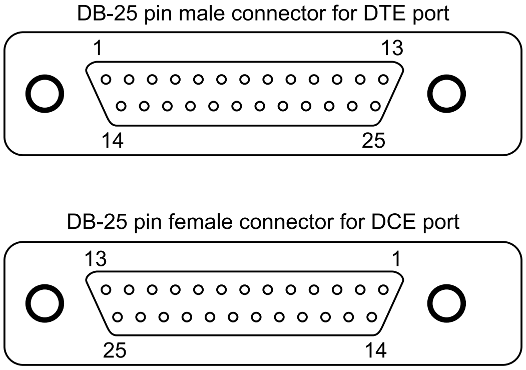

DB-25 pin male connector for DTE port
1
13
14
25
DB-25 pin female connector for DCE port
13
1
25
14

Fig. 6.40 Pin connector of EIA-232-D interface

Electrical Specifications

The electrical specifications of the EIA-232-D interface specify characteristics of the electrical signals. EIA-232-D is a voltage interface. Positive and negative voltages within the limits as shown in Figure 6.41 are assigned to the two logical states of a binary digital signal.

Limit .... + 25 volts

Logic 0, on, Space

Nominal .... + 12 volts

.... + 3 volts

0 volts ....

.... – 3 volts

Nominal .... – 12 volts

Logic 1, off, Mark

Limit .... – 25 volts

Fig. 6.41 Electrical specifications of EIA-232-D interface

All the voltages are measured with respect to the common ground. The 25-volts limit is the open circuit or no-load voltage. The range from – 3 to + 3 volts is the transition region and is not assigned any state.

DC resistance of the load impedance is specified to be between 3000 to 7000 ohms with a shunt capacitance less than 2500 PF. The cable interconnecting a DTE and a DCE usually has a capacitance of the order of 150 PF/m which limits its maximum length to about 16 metres. EIA-232-D specifies the maximum length of the cable as 50 feet at the maximum data rate of 20 Kbps.

Functional Specifications

Functional specifications describe the various signals which appear on different pins of the EIA-232-D interface. Figure 6.42 lists these signals which are divided into five categories.

Pin	To	To	Circuit names	CCITT	EIA
	DTC	DCE			
1	Common		Shield	101	—
7	Common		Signal ground	107	AB
2		→	Transmitted data	103	BA
3	←		Received data	104	BB
4		→	Request to send	105	CA
5	←		Clear to send	106	CB
6	←		DCE ready	107	CC
20		→	Data terminal ready	108.2	CD
22	←		Ring indicator	125	CE
8	←		Received line signal detector	109	CF
21	←		Signal quality detector	110	CG
23		→	Data rate selector (DTE)	111	CH
23*	←		Data rate selector (DCE)	112	CI
24		→	Transmitter signal element timing (DTE)	113	DA
15	←		Transmitter signal element timing (DCE)	114	DB
17	←		Receiver signal element timing (DCE)	115	DD
14		→	Secondary transmitted data	118	SBA
16	←		Secondary received data	119	SBB
19		→	Secondary request to send	120	SCA
13	←		Secondary clear to send	121	SCB
12*	←		Secondary received line signal detector	122	SCF
18		→	Local loopback	141	LL
21		→	Remote loopback	140	RL
25	←		Test mode	142	TM

Fig. 6.42 EIA-232-D interchange circuits

1. Ground or common return
2. Data circuits
3. Control circuits
4. Timing circuits
5. Secondary channel circuits

Procedural Specifications

Procedural specifications lay down the procedures for the exchange of control signals between a DTE and a DCE. Following are the sequence of events which comprise the complete procedure for data transmission.

1. Equipment readiness phase
2. Circuit assurance phase
3. Data transfer phase
4. Disconnect phase

Equipment Readiness Phase

During the equipment readiness phase following functions are carried out.

- The DTE and DCE are energized.
- Physical connection between the DCEs is established if the are connected to PSTN.
- The transmission medium is connected to the DCE electronics.
- The DCEs and DTE exchange signals which indicate their ready state.

Circuit Assurance Phase

In the circuit assurance phase, the DTEs indicate their intent to transmit data to the respective DCEs and the end-to-end (DTE to DTE) data circuit is activated. If the transmission mode is half duplex, only one of the two directions of transmission of the data circuit is activated.

Data Transfer Phase

Once the circuit assurance phase is over, data exchange between DTEs can start. The following circuits are in **ON** state during this phase.

Transmitting End	**Receiving End**
Data Terminal Ready	Data Terminal Ready
DCE Ready	DCE Ready
Request to Send	Received Line Signal Detector
Clear to Send	

Disconnect Phase

After the data transfer phase, disconnection of the transmission media is initiated by a DTE. It withdraws Data Terminal Ready Signal. The DCE disconnects from the transmission media and turns off the DCE Ready Signal.

6.8 MODEMS

6.8.1 Introduction

The term 'Modem' is derived from the words, modulator and demodulator. A modem contains a modulator as well as a demodulator. A typical data connect set up using modems is shown in Figure 6.43.

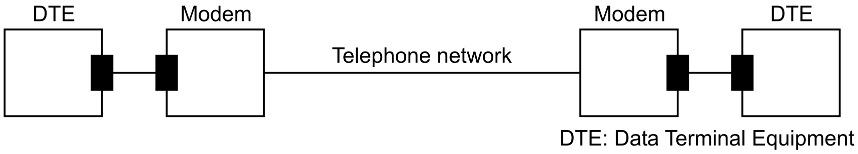


Fig. 6.43 A Data circuit implemented using modems

The digital terminal devices which exchange digital signals are called Data Terminal Equipment (DTE). Two modems are always required, one at each end. The modem at the transmitting end converts the digital signal from the DTE into an analog signal by modulating a carrier. The modem at the receiving end demodulates the carrier and hands over the demodulated digital signal to the DTE.

The transmission medium between the two modems can be dedicated leased circuit or a switched telephone circuit. In the latter case, modems are connected to the local telephone exchange. Whenever data transmission is required, connection between the modems is established through the telephone exchanges. Modems are also required within a building to connect terminals which are located at distances usually more than 15 metres from the host.

Broadly, a modem comprises a transmitter, a receiver and two interfaces as shown in Figure 6.44. The digital signal to be transmitted is applied to the transmitter. The modulated

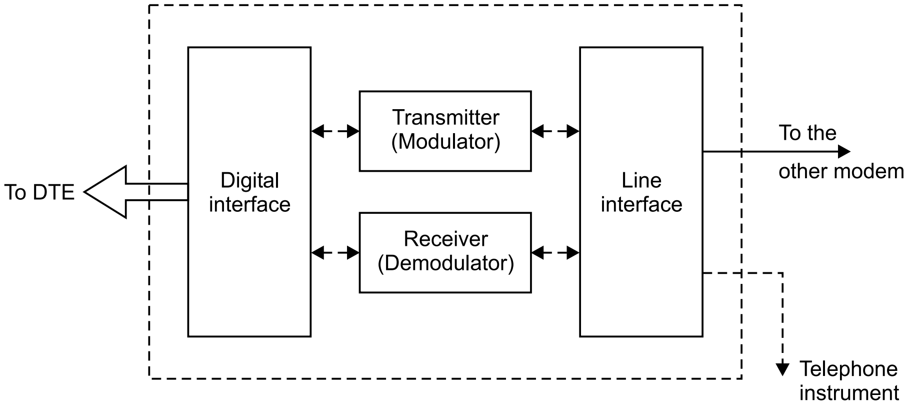


Fig. 6.44 Building blocks of a modem

carrier which is received from the distant end is applied to the receiver. The digital interface connects the modem to the DTE which generates and receives the digital signals. The line interface connects the modem to the transmission channel for transmitting and receiving the modulated signals. Modems connected to telephone exchanges have additional provision for connecting a telephone instrument. The telephone instrument enables establishment of the telephone connection.

The transmitter and receive in a modem comprise several signal processing circuits which include a modulator in the transmitter and a demodulator in the receiver.

6.8.2 Types of Modems

Modems are of several types and they can be categorized in a number of ways based on following basic modem features; as shown in Figure 6.45.

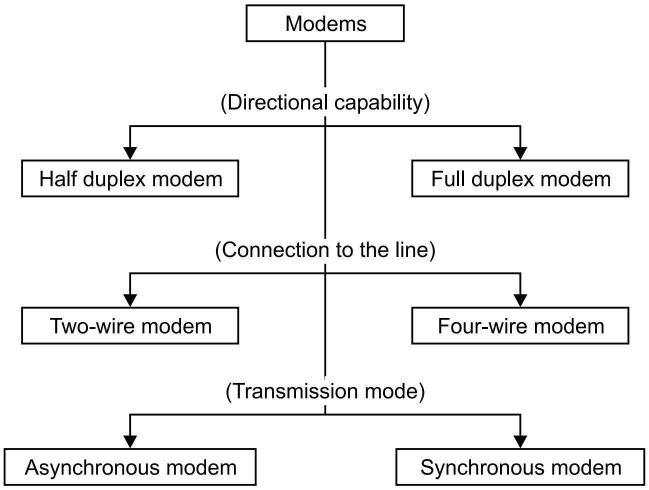


Fig. 6.45 Types of modems

- Directional capability—Half duplex and full duplex modem
- Connection to the line—2-wire modem and 4-wire modem
- Transmission mode—Asynchronous and synchronous modem

Half Duplex and Full Duplex Modems

Half Duplex Modem

A half duplex modem permits transmission in one direction at a time. If a carrier is detected on the line by the modem, it gives an indication of the incoming carrier to the DTE through a control signal of its digital interface as shown in Figure 6.46. So long as the carrier is being received, the modem does not give clearance to the DTE to transmit.

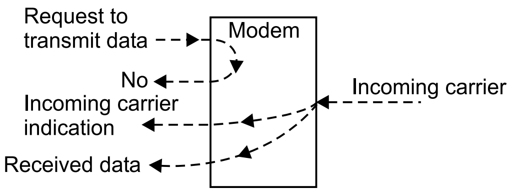


Fig. 6.46 Half duplex modem

Full Duplex Modem

A full duplex modem allows simultaneous transmission in both directions. Thus, there are two carriers on the line, one outgoing and the other incoming as shown in Figure 6.47.

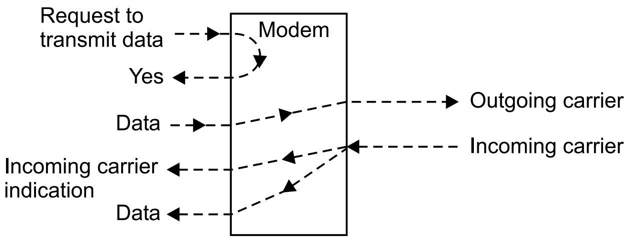


Fig. 6.47 Full duplex modem

2-Wire and 4-Wire Modems

2-Wire Modem

In a 2-wire connection only one pair of wires is extended to the subscriber's premises. In this modems with a 2-wire line interface are required. Such modem use the same pair of wires for outgoing and incoming carriers. Half duplex mode of transmission using the same frequency for the incoming and outgoing carriers, as shown in Figure 6.48 (a). The transmit and receive carrier frequencies can be the same because only one of them is present on the line at a time.

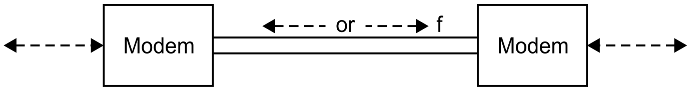


Fig. 6.48 (a) 2-Wire half duplex modem

For full duplex mode of operation on a 2-wire connection, it is necessary to have two transmission channels, one for the transmit direction and the other for the receive direction as shown in Figure 6.48 (b).

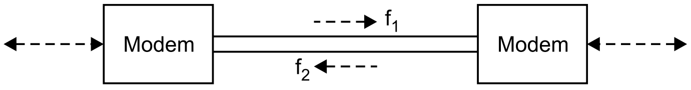


Fig. 6.48 (b) 2-Wire full duplex modem

This is achieved by frequency division multiplexing of two different carrier frequencies. These carriers are placed within the bandwidth of the speech channel. A modem transmits data on one carrier and receives data from the other end on the other carrier. A hybrid is provided in the 2-wire modem to couple the line to its modulator and demodulator as shown in Figure 6.49. There is a special technique which allows simultaneous transmission of incoming and outgoing carriers having the same frequency on the 2-wire transmission medium. Full bandwidth of the speech channel is available to both the carriers simultaneously. This technique is called **echo cancellation technique**.

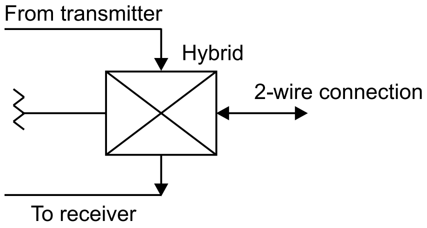


Fig. 6.49 Line Interconnection in a 2-wire full duplex modem

4-Wire Modem

In a 4-wire connection, one pair of wires is used for outgoing carrier and the other is used for the incoming carrier as shown in Figure 6.50. Full duplex and half-duplex modes of data transmission are possible on a 4-wire connection.

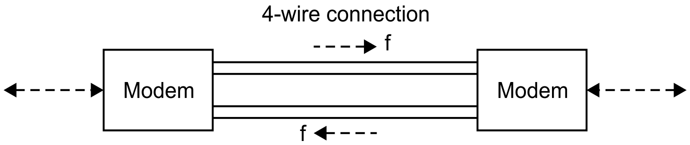


Fig. 6.50 4-Wire modem

Asynchronous and Synchronous Modems

Asynchronous Modem

An asynchronous modem can only handle data bytes with start and stop bits. There is no separate timing signal or clock between the modem and the DTE, as shown in Figure 6.51. The internal timing pulses are synchronized repeatedly to the leading edge of the start pulse.

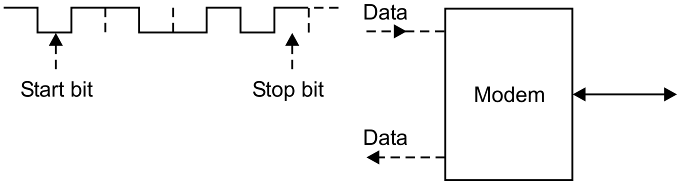
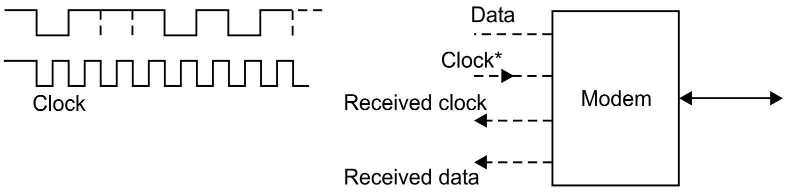


Fig. 6.51 Asynchronous modem

Synchronous Modem

A synchronous modem can handle a continuous stream of data bits but requires a clock signal as shown in Figure 6.52. The data bits are always synchronized to the clock signal. There are separate clocks for the data bits being transmitted and received.

For synchronous transmission of data bits, the DTE can use its internal clock and supply the same to the modem. Else, it can take the clock from the modem and send data bits on each occurrence of the clock pulse. At the receiving end, the modem recovers the clock signal from the received data signal and supplies it to the DTE. High speed modems are provided with scramblers and descramblers for this purpose.



*Transmit clock is supplied either by the modem or by the DTE

Fig. 6.52 Synchronous modem

Scrambler and Descrambler

Scrambler

A scrambler is incorporated in the modems which operate at data rates of 4800 bps and above. The data stream received from the DTE at the digital interface is applied to the scrambler. The scrambler divides the data stream by the generating polynomial and its output is applied to the encoder.

The scrambler at the transmitter consists of a shift register with some feedback loops and exclusive OR gates. Figure 6.53 shows a scrambler used in the CCITT V.27 4800 bps modem.

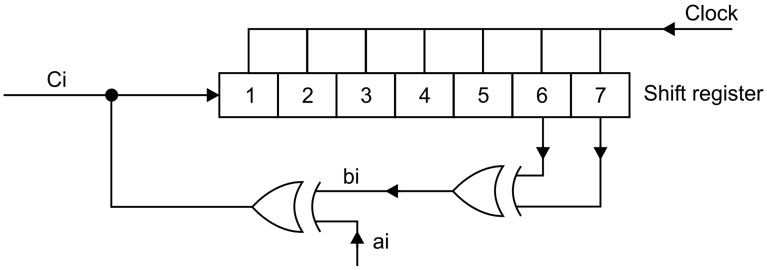


Fig. 6.53 Scrambler used in CCITT V.27 modem

Descrambler

The decoder output is applied to the descrambler which multiplies the decoder output by the generating polynomial. The unscrambled data is given to the DTE through the digital interface. Descrambler is shown in Figure 6.54.

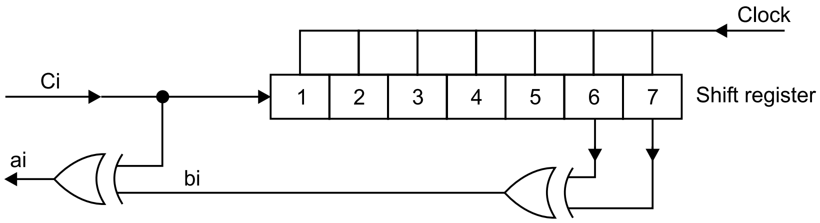


Fig. 6.54 Descrambler used in CCITT V.27 modem

6.8.3 Block Schematic of a Modem

The modem design and complexity vary depending on the bit rate, type of modulation and other basic features.

Figure 6.55 shows important components of a typical synchronous differential PSK modem.

Digital Interface

The digital interface connects the internal circuits of the modem to the DTE. On the DTE side, it consists of several wires carrying different signals. These signals are either from the DTE or from the modem. The digital interface contains drivers and receivers for these signals. The signals are like TD (Transmitted Data), RD (Received Data), DTR, DSR, RTS, CTS, etc.

The most common standard digital interface is EIA-232-D.

Scrambler

The data stream received from the DTE at the digital interface is applied to the scrambler. The scrambler divides the data stream by the generating polynomial and its output is applied to the encoder.

Encoder

An encoder consists of a serial to parallel converter for grouping the serial data bits received from the scrambler. The data bit groups are then encoded for different PSK.

Modulator

A modulator changes the carrier phase as per the output of the encoder. A pulse shaping filter precedes the modulator to reduce the intersymbol interference. Raised cosine pulse shape is usually used.

Compromise Equalizer

It is a fixed equalizer which provides pre-equalization of the anticipated gain and delay characteristics of the line.

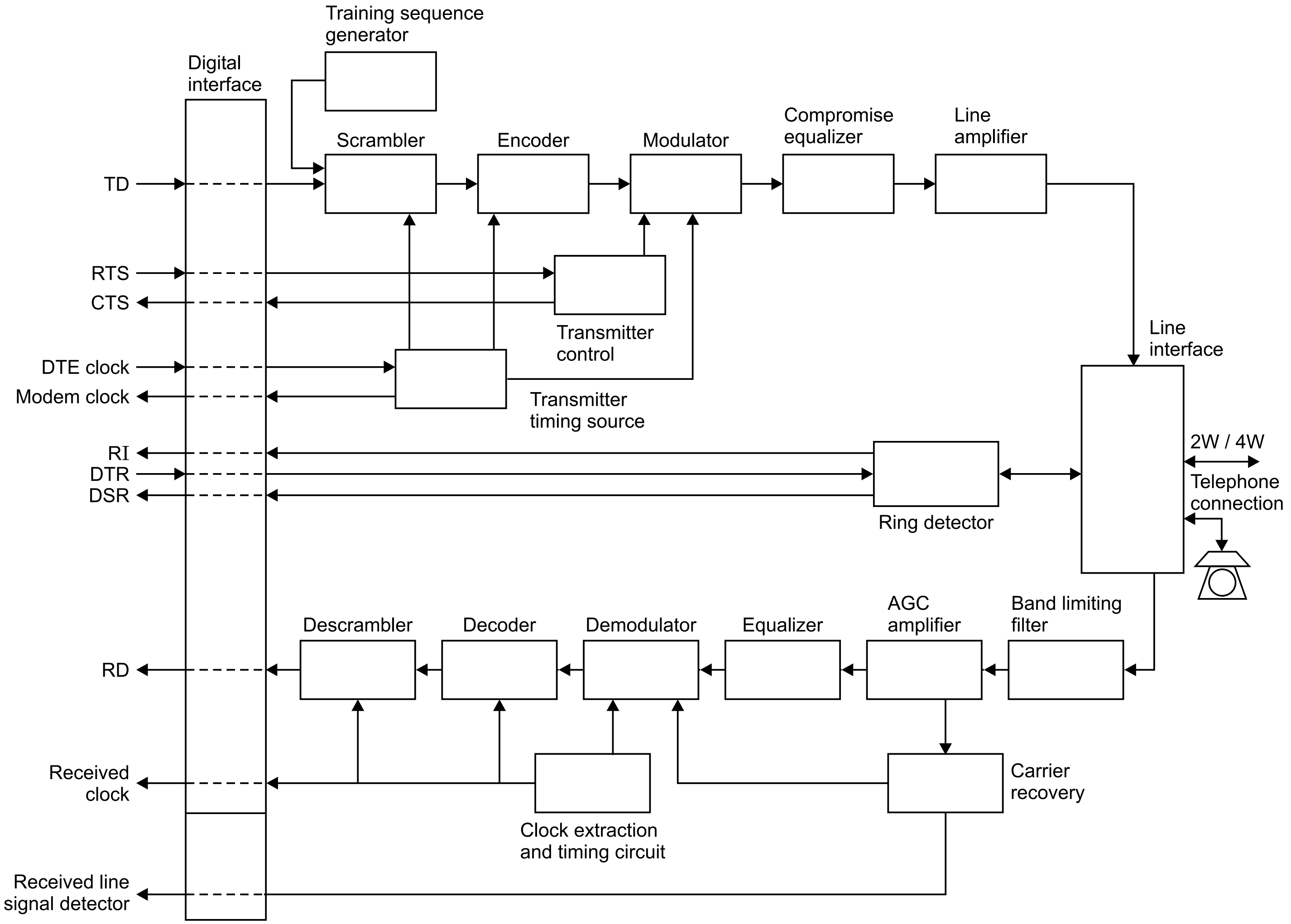


Fig. 6.55 Components of a differential PSK modem

Line Amplifier

The line amplifier is provided to bring the carrier level to the desired transmission level output of the line amplifier is coupled to the line through the line interface.

Transmitter Timing Source

Synchronous modems have an in-built crystal clock source which generates all the timing references required for the operation of the encoder and the modulator. The clock is also supplied to the DTE through the digital interface. The modem has provision to accept the external clock supplied by the DTE.

Transmitter Control

This circuit controls the carrier transmitted by the modem. When the RTS is received from the DTE, it switches on the outgoing carrier and send it on the line. After a brief delay, it generates the CTS signal for the DTE so that it may start transmitting data.

Training Sequence Generator

The data signals reception through the modem requires following operational conditions established in the receiver portion of the modem.

- The demodulator carrier is detected and recovered.
- The adaptive equalizer is conditioned for the line characteristics.
- The receiver timing clock is synchronized.
- The descrambler is synchronized to the scrambler.

These functions are carried out by sending a training sequence.

Line Interface

The line interface provides connection to the transmission facilities through coupling transformers.

Receive Band Limiting Filter

In the receive direction, the band limiting filter selects the received carrier from the signals present on the line. It also removes the out-of-band noise.

AGC Amplifier

Automatic Gain Control (AGC) amplifier provides variable gain to compensate for carrier-level loss during transmission. The gain depends on the received carrier level.

Equalizer

The equalizer section of the receiver corrects the attenuation and group delay distortion introduced by the transmission medium and the band limiting filters. Fixed, manually adjustable or adaptive equalizers are provided depending on speed, line condition and the application.

Carrier Recovery Circuit

The carrier is recovered from the AGC amplifier output by this circuit. The recovered carrier is supplied to the demodulator. An indication of the incoming carrier is given at the digital interface.

Demodulator

The demodulator recovers the digital signal from the received modulated carrier.

The carrier required for demodulation is supplied by the carrier recovery circuit.

Clock Extraction Circuit

The clock extraction circuit recovers the clock from the received digital signal.

Decoder

The decoder performs a function complementary to the encoder. The demodulated data bits are converted into groups of data bits which are serialized by using a parallel to serial converter.

Descrambler

The decoder output is applied to the descrambler which multiplies the decoder output by the generating polynomial. The unscrambled data is given to the DTE through the digital interface.

QUESTIONNARIES

1. Explain Design issues of physical layer.
2. Explain the term channel capacity? Explain the design issues for a physical layer.
3. Explain the following types of guided media:
   - Optical fiber
   - Twisted pair
   - Co-axial cable
4. Describe the following term related to unguided media:
   - Cellular telephone
   - RF allocation
5. Explain EIA-232 digital interface.
6. Draw block diagram of MODEM and explain.
7. Explain various types of MODEMS and specify need of scrambler in the same.
8. Explain different MODEM standards.
9. Explain EIA-232-D interface specifications.
10. Write short note on:
    - Modem
    - Modems and types of Modems.

Chapter 7

DATA LINK LAYER

INTRODUCTION

In this chapter we will study the design of layer 2, the data link layer of OSI model. This study deals with the algorithms for achieving reliable, efficient communication between two adjacent machines at the data link layer.

First you think about the problem that there is no software to study communication. Unfortunately, communication circuits make errors occasionally. Furthermore, they have only a finite data rate and there is a nonzero propagation delay between the time a bit is sent and the time it is received. These limitations have important implications for the efficiency of the data transfer. The protocols used for communications must take all these factors into consideration. These protocols are the subject of this chapter.

7.1 DATA LINK LAYER DESIGN ISSUES

The data link layer has a number of specific functions to carry out. These functions include providing a well-defined service interface to the network layer, determining how the bits of the physical layer are grouped into frames, dealing with transmission errors and regulating the flow or frames so that slow receivers are not swamped by fast senders. In the following sections we will examine each of these issues in turn.

7.2 SERVICES PROVIDED TO THE NETWORK LAYER

The function of the data link layer is to provide services to the network layer. The principal service is transferring data from the network layer on the source machine to the network layer on the destination machine. On the source machine there is an entity, call it a process, in the network layer that hands some bits to the data link layer for transmission to the destination. The job of the data link layer is to transmit the bits to the destination machine, so they can be handed over to the network layer there, as shown in Figure 7.1 (a).

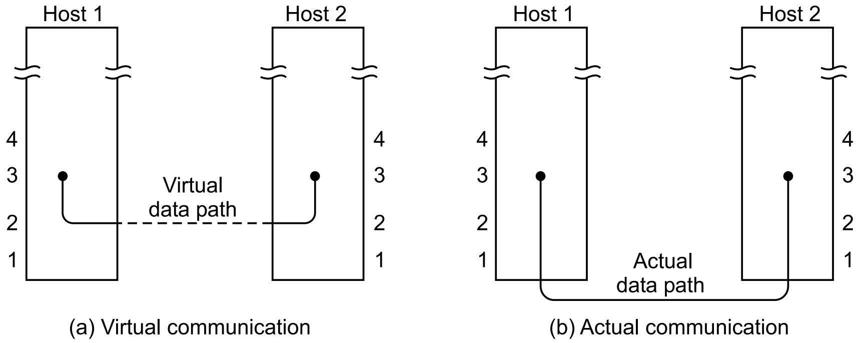


Fig. 7.1 (a) Virtual communication (b) Actual communication

The actual transmission follows the path of Figure 7.1 (b), but it is easier to think in terms of two data layer processes communicating using a data link protocol.

The data link layer can be designed to offer various services. The actual services offered can vary from system to system. Three reasonable possibilities that are commonly provided are:

1. Unacknowledged connectionless service
2. Acknowledged connectionless service
3. Acknowledged connection-oriented service

Unacknowledged connectionless service consists of having the source machine send independent frames to the destination machine without having the destination machine acknowledge them. No connection is established beforehand or released afterward. If a frame is lost due to noise on the line, no attempt is made to recover it in the data link layer. This class of service is appropriate when the error rate is very low so recovery is left to higher layers. It is also appropriate for real-time traffic, such as speech, in which late data are worse than bad data. Most LANs use unacknowledged connectionless service in the data link layer.

The next step up in terms of reliability is acknowledged connectionless service. When this service is offered, there are still no connections used, but each frame sent is individually acknowledged. In this way, the sender knows whether or not a frame has arrived safely. If it has not arrived within a specified time interval, it can be sent again. This service is useful over unreliable channels, such as wireless systems.

It is perhaps worth emphasizing that providing acknowledgements in the data link layer is just an optimization, never a requirement. The transport layer can always send a message and wait for it to be acknowledged. If the acknowledgement is not forthcoming before the timer goes off, the sender can just send the entire message again. The trouble with this strategy is that if the average message is broken up into, say 10 frames, and 20 per cent of all frames are lost, it may take a very long time for the message to get through. If individual frames are acknowledged and retransmitted, entire message get through much faster. On reliable channels, it is well worth the cost due to their inherent unreliability.

Getting back to our services, the most sophisticated service the data link layer can provide to the network layer is connection-oriented service. With this service, the source and destination machines establish a connection before any data link layer guarantees that each frame sent is indeed received. Furthermore, it guarantees that each frame is received exactly one and that all frames are received in the right order. With connectionless service, in contrast, it is conceivable that a lost acknowledgement cause a frame to be sent several times and thus received several times. Connection-oriented service, in contrast, provides the network layer processes with the equivalent of a reliable bit stream.

When connection-oriented service is used, transfers have three distinct phases. In the first phase the connection is established by having both sides initialize variables and counters needed to keep track of which frames have been received and which ones have not. In the second phase, the connection is released, freeing up the variables, buffers, and other resources used to maintain the connection.

Take an example of say a WAN subnet consisting of routers connected by point-to-point leased telephone lines. When a frame arrives at a router, the hardware verifies the checksum and passes the frame to the data link layer software. The data link layer software checks to see if this is the frame expected, and if so, gives the packet contained in the payload field to the routing software. The routing software chooses the appropriate outgoing line and passes the packet back down to the data link layer software, which then transmits it. The flow over two routers is shown in Figure 7.2.

The routing code frequently wants the job done right, that is, reliable, sequenced connections on each of the point-to-point lines. It does not want to be bothered too often with packets that got lost on the way. It is up to the data link protocol, shown in the dotted rectangle, to make unreliable communication lines look perfect, or at least, fairly good. This property is especially important for wireless links, which are inherently very unreliable. As an aside, although we have shown multiple copies of the data link layer software in each router, infact, one copy handles all the lines, with different tables and data structures for each one.

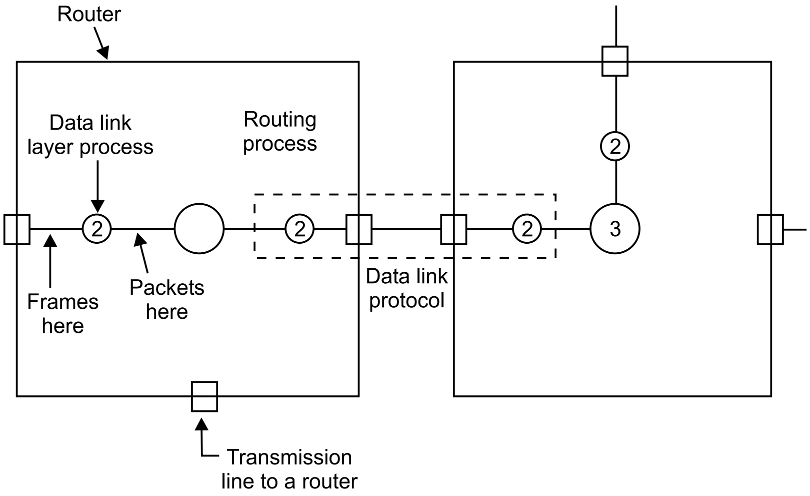


Fig. 7.2 Placement of the data link protocol

7.3 FRAMING METHODS

Before discussing the different framing techniques, it is pertinent to make **two** important observations:

1. In many subnets, especially in WANs, the frame size is chosen less than the packet size for achieving high reliability and efficiency of the DLC layer communication. Thus, a packet to be transmitted is often fragmented into multiple frames each of which is properly framed. (delimited) and sent to the other side where all the received frames are properly reassembled to reconstruct the original packet, ready to be handed over to the destination.
2. The present discussions relate to framing in the context of synchronous communication. For asynchronous communication, *e.g.* the one-character-at-a-time communication between a dump terminal and a host computer, the data to be transferred is delimited by a START bit and a STOP bit. This delimiting allows the receiver not only to identify the start and end of the 1-byte long "miniframe" but also to synchronize with the transmitter. However, this synchronisation aspect is unnecessary for synchronous communication because the receiver is always in synchronization.

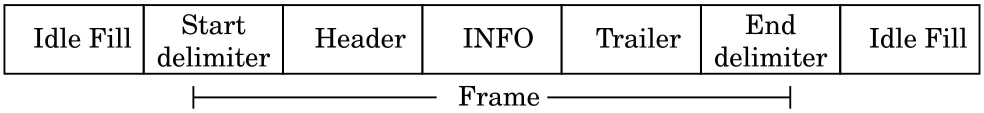


Fig. 7.3 General format of frame

The general format of a frame is shown in Figure 7.3 in which the variable length INFO field contains either the entire packet handed over by the network layer sending process to the DLC layer sending process or a part of it. The subject of discussion in the present section is the various techniques of framing, *i.e.*, the choice of the frame delimiters.

The following three methods are commonly employed for framing:

1. Character-Oriented Framing
2. Bit-Oriented Framing
3. Code violation-Oriented Framing.

1. Character-Oriented Framing

This was one of the earliest schemes for delimiting packets containing character data, and it employs three special control characters in ASCII viz., STX (Start of Text), ETX (End of Text) and DLE (Data Link Escape), for the purpose of framing. Assuming that two independent character sequences "MAY" and "2000" are to be transmitted with some intervening pause, the data actually transmitted by the DLC layer sending process is:

SYN SYN STX MAY ETX SYN SYN STX 2000 ETX SYN SYN

Where the control character SYN (SYNchronous idle) is the idling character for the link. It may be observed that if the data to the transferred is not just a sequence of data characters but any general binary data (*e.g.* object code, disk image, graphical data, etc.), the delimiting special control characters may also occur in the data and may cause confusion at the receiver. For accommodating general binary data, an escape mode was created with the help of the special control character DLE (Data Link Escape) *i.e.*, the delimiting was now done by the character pairs DLE STX and DLE ETX. However, this modification only reduced, although significantly, the probability of occurrence of the delimiting patterns in the data. For eliminating the problem altogether, a new idea called "character stuffing" emerged. According to the character-stuffing algorithm, the transmitter always doubles the DLEs in the data. *i.e.*, whenever the transmitter observes a DLE character in the data stream, it stuffs an additional DLE character just before it so that the delimiting character patterns DLE STX and DLE ETX do not occur in the transmitted data stream. The receiver, on its part, simply removes (*i.e.*, destuffs) the first DLE character whenever it sees the occurrence of a double DLE. The principle of character stuffing is illustrated in Figure 7.4 (a) and it achieves complete data transparency between the DLC layer (service provider) and the network layer (service user).

* SYN SYN DLE STX ETX DLE ETX STX DLE DLE ETX SYN SYN

Data before character stuffing

Frame without character stuffing

* SYN SYN DLE STX ETX DLE DLE ETX STX DLE DLE DLE ETX SYN SYN

Data after character stuffing

Frame with chara1cter stuffing

(*a*) Character stuffing

* 11111111 0 111111 0 0 11 0 11111 0 1111111 0 0 111111 0 1111111

Data before Bit stuffing

Frame without Bit stuffing

* 11111111 0 11111 0 1 0 11111 0 0 11111 0 11 0 1111 0 0 111111 0 1111111

Data bit after Bit stuffing

Frame with Bit stuffing

(*b*) Bit stuffing

Fig. 7.4 Data stuffing

2. Bit-Oriented Framing

In bit-oriented framing, delimiting is done by the special bit patterns "01111110" called a "flag". For achieving data transparency as explained earlier in the case of character stuffing, whenever the transmitter sees a string of 5 consecutive 1's occurring in the data stream (in bit-oriented framing, an arbitrary data stream is viewed simply as a bit stream), it stuffs a 0 after the string. Whenever the receive finds a 0 following 5 consecutive 1's, it recognises it as a stuffed bit and just removes it. Figure 7.4 (b) illustrates the concept of bit stuffing for achieving data transparency in bit-oriented synchronous data communication. It should be noted that the bit-oriented framing has less overhead compared to character oriented framing. And, additionally, allows any arbitrary number of bits in the data field rather than restricting the data field to be only a stream of 8-bit characters. However, so far as common data communication is concerned, data is usually organised as bytes.

3. Code Violation-Oriented Framing

Instead of using special character or bit patterns at the virtual bit pipe level, it is possible to do framing at the real bit pipe level while physically encoding bits into electrical signals. Let us consider for example, the Manchester coding, which is a baseband encoding technique that represents a 1 as a high-low pair and a 0 as a low-high pair. The basic feature of this code is that there is always a transition in the middle of the signalling interval, whenever a data-bit is being transmitted. Now, frame delimiting can be done through violation of this basic feature of the code by sending, say, one high-high pair followed by a low-low pair. This ingenious scheme avoids the expected transition in the middle of the bit interval and thereby indicates the frame boundaries to the receiver. This framing technique of using an invalid physical layer coding has been employed in the 802 LAN standards.

Before concluding this section on framing, it may be appropriate to point out that sending in the frame header the count of the number of characters or bits present in the frame being transmitted has also been used to let the receiver determine the end of the frame (end delimiting). However, this method is no more used because bit errors can affect the count lead to serious. Synchronisation problem at the receiver. Of course, if the count is used in conjunction with one of the three framing techniques described above, it can help in increasing the reliability of the framing process significantly.

7.4 ERROR CONTROL-DETECTION AND CORRECTION

Error Control

Error control refers to mechanisms to correct and detect errors that occur in the transmission of frames. The model that we will use, can shown in Figure 7.5. As before, data are sent as a sequence of frames:

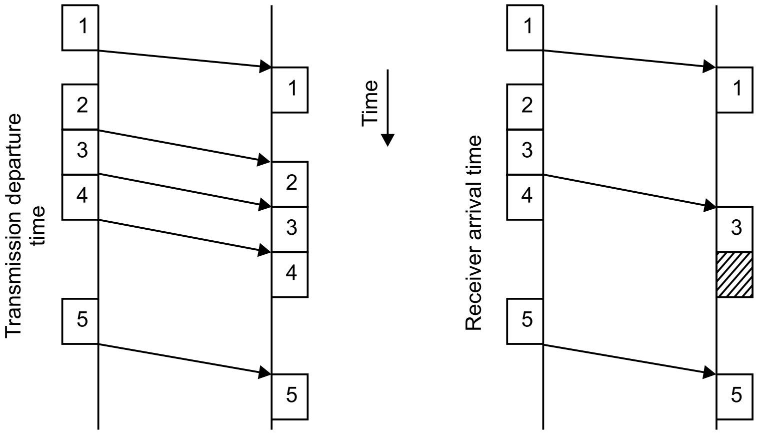


Fig. 7.5 Model of frame transmission

Frames arrive in the same order in which they are sent, and each transmitted frame suffers an arbitrary and variable amount of delay before reception. In addition, we admit the possibility of two types of errors.

- **Lost frame:** A frame fails to arrive at the other side. For example, a noise burst may damage a frame to the extent that the receiver is not aware that a frame has been transmitted.
- **Damaged frame:** A recognizable frame does arrive, but some of the bits are in error (have been altered during transmission).

The most common techniques for error control are based on following ingredients.

- **Error detection:** As described in the next section.
- **Positive acknowledgement:** The destination returns a positive acknowledgement to successfully received, error-free frames.
- **Retransmission after time out:** The source retransmits a frame that has not been acknowledged after a predetermined amount of time.
- **Negative acknowledgement and retransmission:** The destination returns a negative acknowledgement to frames in which an error is detected. The source retransmits such frames.

Collectively, these mechanisms are all referred to as automatic repeat request (ARQ). The effect of ARQ is to turn an unreliable data link into a reliable one.

Error-Correction

To understand how errors can be handles, it is necessary to look closely at what an error really is. Normally, a frame consists of *m* data (*i.e.*, message) bits and r redundant, or check bits.

Let the total length be *n* (*i.e.*, $n = m + r$). An n-bit, unit containing data and checkbits is often referred to as an n-bit codeword.

Given any two codewords, say, 10001001 and 1011001, it is possible to determine how many corresponding bit differ. In this case, 3 bits differ. To count the number of 1 bits in the result. The number of bit positions in which two codewords differ is called the **Hamming distance** (Hamming, 1950). Its significance is that if two codewords are a Hamming distance d apart, it will require d single-bit errors to convert one into the other.

In most data transmission applications, all 2^m possible data messages are legal, but due to the way the check bits are computed, not all the 2^n possible codewords are used. Given the algorithm for computing the check bits, it is possible to constract a complete list of the legal codewords, and from this list find the two codewords whose, Hamming distance is minimum. This distance is the Hamming distance of the complete code.

The error-detecting and error-correcting properties of a code depend on its Hamming distance. To detect d errors, you need a distance $d + 1$ code because with such a code there is no way that d single-bit errors can change a valid codeword into another valid codeword. When the receiver sees an invalid codeword, it can tell that a transmission error has occurred. Similarly, to correct *d* errors, you need a distance 2*d* + 1 code because that way the legal codewords are so far apart that even with *d* changes, the original codeword is still closer than any other codeword, so to can be uniquely determined.

Consider a code in which a single parity bit is appended to the data. The parity bits is chosen so that the number of 1 bits in the codeword is even (or odd).

Example for an error-correcting code: Consider a code with only four valid codewords:

0000000000, 0000011111, 1111100000, and 1111111111

This code has a distance 5, which means that it can correct double errors. If the codeword 0000000111 arrives, the receiver knows that the original must have been 0000011111. If, however, a triple error changes 0000000000 into 0000000111, the error will not be corrected property.

Imagine that we want to design a code with *m* message bits and *r* check bits that will allow all single errors to be corrected. Each of the 2^m legal messages has *n* illegal codewords at a distance 1 from it. These are formed by systematically inverting each of the *n* bits in the *n*-bit codeword formed from it. Thus each of the 2^m legal messages requires $n + 1$ bit patterns dedicated to it. Since the total number of bit patterns is 2^n, we must have $(n + 1)2^m \leq 2^n$. Using $n = m + r$, this requirement becomes $(m + r + 1)\ 2^r$. Given m, this puts a lower limit on the number of check bits needed to correct single errors.

This theoretical lower limit can, in fact, be achieved using a method due to Hamming. The bits of the codeword are numbered consecutively, starting with bit 1 at the left end. The bits that are powers of 2 (1, 2, 4, 8, 16, etc.) are check bits. The rest (3, 5, 6, 7, 9, etc.) are filled up with the *m* data bits. Each check bit forces the parity of some collection of bits, including itself, to be even (or odd). A bit may be included in several parity computations. To see which check bits the data bit in position *k* contributes to, rewrite *k* as a sum of powers or 2 for example, 11 = 1 + 2 + 8 and 29 = 1 + 4 + 8 + 16. A bit is checked by just those check bits occurring in its expansion (*e.g.*, bit 11 is checked by bits 1, 2 and 8).

When a codeword arrives, the receiver initializes a counter to zero. It then examines each check bit, *k* (*k* = 1, 2, 4, 8, ...) to see if it is has the correct parity. If not, it adds k to the counter. If the counter is zero after all the check bits it have been examined. (*i.e.*, if they were all correct), the codeword is accepted as valid. If the counter is nonzero, it contains the number of the incorrect bit. For example, if check bits 1, 2, and 8 are in error, the inverted bits is 11, because it is the only one checked by its 1, 2, and 8. Figure 7.6 shows some *y*-bit ASCII characters encoded as 11-bit codewords using a Hamming code. Remember that the data is found in bit positions 3, 5, 6, 7, 9, 10 and 11.

Hamming codes can only correct single errors. However, there is a trick that can be used to permit Hamming codes to correct burst errors. A sequence of *k* consecutive codewords are arranged as a matrix, one codeword per row. Normally, the data would be transmitted one codeword at a time, from left to right. To correct burst errors, the data should be transmitted one column at a time, starting with the leftmost column. When all *k* bits have been sent, the second column is sent, and so on. When the frame arrives at the receiver, the matrix is reconstructed, one column at a time. If a burst error of length *k* occurs, at most 1 bit in each of the k codewords will have been affected, but the Hamming code can correct one error per codeword, so the entire block can be restored. This method uses k_r check bits to make block of k_m data bits immune to a single burst error of length k or less.

Char.	**ASCII**	**Check bits**
H	1001000	00110010000
a	1100001	10111001001
m	1101101	11101010101
m	1101101	11101010101
i	1101000	01101011001
n	1101110	01101010110
g	1100111	11111001111
	0100000	10011000000
c	1100011	11111000011
o	1101111	00101011111
d	1100100	11111001100
e	1100101	00111000101

Order of bit transmission

Fig. 7.6 Use of a hamming code to correct burst errors

Error Detection

Regardless of the design of the transmission system, there will be error, resulting in the change of one or more bits in a transmitted frame.

Let us define these probabilities with respect to errors in transmitted frames.

P_b: Probability of a single bit error, also known as the bit error rate.

P_1: Probability that a frame arrives with no bits errors.

P_2: Probability that a frame arrives with one or more undetected bit errors.

P_3: Probability that a frame arrives with one or more detected bit errors but no undetected bit errors.

First, consider the case when no means are taken to detect errors; the probability of detected errors (P_3) then, is zero. To express the remaining probabilities, assume that the probability that any bit is in error (P_b) is constant and independent for each bit. Then we have

$$P_1 = (1 - P_b)^F$$
$$P_2 = 1 - P_1$$

where F is the number of bits per frame. In words, the probability that a frame arrives with no bit errors decreases when the probability of a single bit error increases, as you would expect. Also, the probability that a frame arrives with no bit errors decreases with increasing frame length, the longer the frame, the more bits it has and the higher the probability that one of these is in error.

Let us take a simple example to illustrate these relationships. A defined object for ISDN connections is that the bit error rate on a 64 Kbps channel should be less than 10^{-6} on at least 90% of observed 1 minute intervals. Suppose now that we have the rather modest user requirement that at most one frame with an undetected bit error should occur per day on a continuously used 64 Kbps channel, and let us assume a frame length of 1000 bits. The number of frames that can be transmitted in a day comes out to 5.529×10^6, which yields a desired frame error rate of $P_2 = 1/(5.529 \times 10^6) = 0.18 \times 10^{-6}$. But, if we assume a value of Pb of 10^{-6}, then $P_1 = (0.999999)^{1000} = 0.999$ and, therefore, $P_2 = 10^{-3}$, which is about three orders of magnitude too large to meet our requirement.

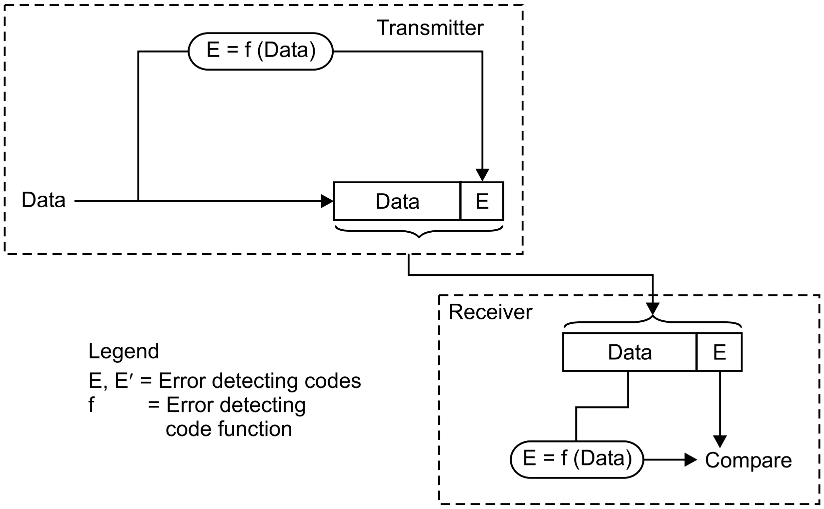


Fig. 7.7 Error detection

This is the kind of result that motivates the use of error-detection techniques. All of these techniques operate on the following principle which is shown in Figure 7.7. For a given frame of bits, additional bits that constitute an error-detecting code are added by the transmitter. This code is calculated as a function of the other transmitted bits. The receiver performs the same calculation and compares the two results. A detected error occurs if and only if there is a mismatch. Thus, P_3 is the probability that if a frame contains errors the error-detection scheme will detect that fact P_2 is known as the **residual error rate,** and is the probability that an error will be undetected despite the use of an error-detection scheme.

Parity Check

The simplest error-detection scheme is to append a parity bit to the end of a block of data. A typical example is ASCII transmission, in which a parity bit is attached to each 7-bit ASCII character. The value of this bit is selected so that the character has an even number of 1s (even parity), or an odd number of 1s (odd parity). So, for example, if the transmitter is transmitting an ASCII G(1110001) and using odd parity, it will append a 1 and transmit 11100011. The receiver examines the received character and, if the total number of 1s is odd, assumes that no error has occurred. If one bit (or any odd number of bits) is erroneously inverted during transmission (for example, 11000011), then the receiver will detect an error. Note, however, that if two (or any even number) of bits are inverted due to error, an undetected error occurs. Typically, even parity is used for synchronous transmission and odd parity for asynchronous transmission.

The use of the parity bit is not fullproof, as noise impulses are often long enough to destroy more than one bit, particularly at high data rates.

Cyclic Redundancy Check (CRC)

One of the most common, and one of the most powerful, error-detecting codes is the cyclic redundancy check (CRC), which can be described as follows. Given a *k*-bit block of bits, or message the transmitter generates an *n*-bit sequence known as a frame check sequence (FCS), so that the resulting frame, consisting of $k + n$ bits, is exactly divisible by some predetermined number. The receiver then divides the incoming frame by that number and, if there is no remainder, assumes there was no error.

For clarification, we present the procedure in three way:

* Modulo 2 arithmetic
* Polynomials
* Digital logic

Modulo 2 Arithmetic : Modulo 2 arithmetic uses binary addition with no carries, which is just the exclusive or operation. For example:

$$\begin{array}{r} 1111 \\ +\ 1010 \\ \hline 0101 \end{array} \qquad \begin{array}{r} 11001 \\ \times\ 11 \\ \hline 11001 \\ 11001 \\ \hline 101011 \end{array}$$

Now define:

T = (*k* + *n*)-bit frame to be transmitted, with *n* < *k*

M = *k*-bit message, the first *k*-bit of T

F = *n*-bit FCS, the last *n*-bits of T

P = Pattern of *n* + 1 bits; this is the predetermined divisor.

We would like T/P to have no remainder. It should be clear that

$$T = 2^n M + F$$

That is, by multiplying M by 2^n, we have, in effect, shifted to be left by *n* bits and padded out the result with zeros. Adding F yields the concatenation of M and F, which is T, we want T to be exactly divisible by P. Suppose that we divided 2^nM by P

$$\frac{2^n M}{P} = Q + \frac{R}{P} \qquad \text{...(7.1)}$$

There is a quotient and a remainder. Because division is binary, the remainder is always at least one bit less than the divisor. We will use this remainder as our FCS. Then,

$$T = 2^n M + R$$

Question: Does this R satisfy our condition that T/P have no remainder? To see that it does, consider

$$\frac{T}{P} = \frac{2^n M + R}{P}$$

Substituting Equation (7.1), we have

$$\frac{T}{P} = Q + \frac{R}{P} + \frac{R}{P}$$

However, any binary number added to itself (modulo 2) yields zero. Thus,

$$\frac{T}{P} = Q + \frac{R + R}{P} = Q$$

There is no remainder, and therefore, T is exactly divisible by P. Thus, the FCS is easily generated. Simply divide 2^nM by P and use the remainder as the FCS. On reception, the receiver will divide T by P and will get no remainder if there have been no errors.

Let us now consider a simple example.

1. Given:

   Message M = 1010001101 (10 bits)

   Pattern P = 110101 (6 bits)

   FCS R = to be calculated (5 bits)

2. The message M is multiplied by 2^5, yielding 101000110100000.

3. This product is divided by P.

   → Equation

$$
\begin{array}{r l}
 & \,11001010011 \leftarrow Q \\
\cline{2-2}
P \rightarrow 110101 & \big|\,101000110100000 \leftarrow 2^n\,M \\
 & \,\underline{110101} \\
 & \,111011 \\
 & \,\underline{110101} \\
 & \,111010 \\
 & \,\underline{110101} \\
 & \,111110 \\
 & \,\underline{110101} \\
 & \,101100 \\
 & \,\underline{110101} \\
 & \,110010 \\
 & \,\underline{110101} \\
 & \,01110 \leftarrow R
\end{array}
$$

4. The remainder (R = 01110) is added to 2^nM to give T = 101000110101110, which is transmitted

5. If there are no errors, the receiver receives T intact. The received frame is divided by P.

   → Equation

$$
\begin{array}{r l}
 & \,11001010011 \leftarrow Q \\
\cline{2-2}
P \rightarrow 110101 & \big|\,101000110101110 \leftarrow 2^n\,M + R \\
 & \,\underline{110101} \\
 & \,111011 \\
 & \,\underline{110101} \\
 & \,111010 \\
 & \,\underline{110101} \\
 & \,111110 \\
 & \,\underline{110101} \\
 & \,101111 \\
 & \,\underline{110101} \\
 & \,110101 \\
 & \,\underline{110101} \\
 & \,000000 \leftarrow R
\end{array}
$$

Because there is no remainder, it is assumed that there have been no errors.

The pattern P is chosen to be one bit longer than the desired FCS, and the exact bit pattern chosen depends on the type of errors expected. At minimum, both the high and low-order bits of P must be 1.

The occurrence of an error is easily expressed. An error results in the reversal of a bit. This is equivalent to taking the exclusive or of the bit and 1 (modulo 2 addition of 1 to the bit) : 0 + 1 = 1; 1 + 1 = 0. Thus, the errors in an ($n + k$)-bit frame can be represented by an ($n + k$)-bit field with 1s in each error position. The resulting frame T_r can be expressed as

$$T_r = T + E$$

Where

T = Transmitted frame

E = error pattern with 1s in position where error occur

T_r = received frame

The receiver will fail to detect an error if an only if T_r is divisible by P. Which is equivalent to E divisible by P. Intuitively, this seems an unlikely occurrence.

Polynomials: A second way of viewing the CRC process is to express all values as polynomials in a dummy variable X, with binary coefficients. The coefficients correspond to the bit in the binary number. Thus, for M = 110011, we have $M(X) = X^5 + X^4 + X + 1$, and, For P = 11001, we have $P(X) = X^4 + X^3 + 1$. Arithmetic operations are again modulo 2. The CRC process can now be described as

$$\frac{X^n M(X)}{P(X)} = Q(X) + \frac{R(X)}{P(X)}$$

$$T(X) = X^n M(X) + R(X)$$

An error E(X) will only be undetectable if it is divisible by P(X). It can be shown [PETE 61] that all of the following errors are not divisible by a suitably chosen P(X) and, hence, are detectable:

- All single-bit errors.
- All double-bit errors, as long as P(X) has at least three 1s.
- Any odd number of errors, as long as P(X) contains a factor (X + 1).
- Any burst error for which the length of the burst is less than the length of the divisor polynomial; *i.e.*, less than or equal to the length of the FCS.
- Most larger burst errors.

In addition, it can be shown that if all errors patterns are considered equally likely, then for a burst error of length $r + 1$, the probability is $1/2^r$, where r is the length of the FCS.

Three versions of P(X) are widely used:

$$\text{CRC-16} = X^{16} + X^{15} + X^{2} + 1$$

$$\text{CRC-CCITT} = X^{16} + X^{12} + X^{5} + 1$$

$$\text{CRC-32} = X^{32} + X^{26} + X + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + 1.$$

Digital Logic: The CRC process can be represented by, and indeed implemented as, a dividing circuit consisting of exclusive or gates and a shift register. The shift register is a string of 1-bit storage devices. Each device has an output line, that indicates the value currently stored, and an input line. At discrete time instants, known as **clock times**, the value in the storage device is replaced by the value indicated by its input line. The entire register is clocked simultaneously, causing a 1-bit shift along the entire register.

The circuit is implemented as follows:

1. The register contains *n*-bits, equal to the length of the FCS.
2. There are up to *n*-exclusive or gates.
3. The presence or absence of a gate corresponds to the presence or absence of a term in the divisor polynomial, P(X).

The architecture of this circuit is best explained by first considering an example, which is illustrated in Figure 7.8. In this example, we have to use

Message M = 1010001101 $M(X) = X^{9} + X^{7} + X^{3} + X^{2} + 1$

Divisor P = 110101 $P(X) = X^{5} + X^{4} + X^{2} + 1$

which were used earlier in the discussion.

Part (a) of the figure shows the register implementation. The process begins with the shift register cleared (all zeros). The message, or dividend, is then entered, one bit at a time, starting with the most significant bit.

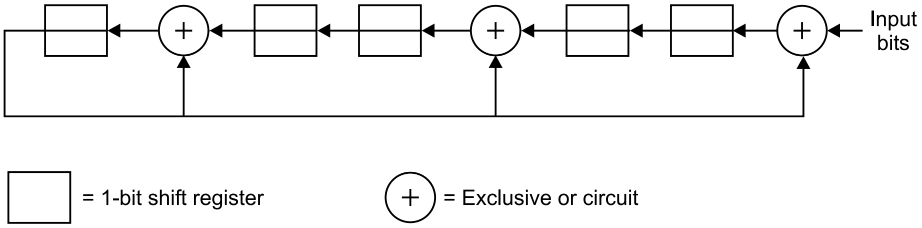


Fig. 7.8 (a) Shift-Register implementation

	C_4	C_3	C_2	C_1	C_0	$C_4 \oplus C_3$	$C_4 \oplus C_1$	$C_4 \oplus$ Input	Input	
Initial	0	0	0	0	0	0	0	1	1	Message to be sent
Step 1	0	0	0	0	1	0	0	0	0	
Step 2	0	0	0	1	0	0	1	1	1	
Step 3	0	0	1	0	1	0	0	0	0	
Step 4	0	1	0	1	0	1	1	0	0	
Step 5	0	1	1	0	0	1	1	1	0	
Step 6	1	1	1	0	1	0	1	0	1	
Step 7	0	1	1	1	0	1	1	1	1	
Step 8	1	1	1	0	1	0	1	1	0	
Step 9	0	1	1	1	1	1	1	1	1	
Step 10	1	1	1	1	1	0	0	1	0	Five zeros added
Step 11	0	1	0	1	1	1	1	0	0	
Step 12	1	0	1	1	0	1	0	1	0	
Step 13	1	1	0	0	1	0	1	1	0	
Step 14	0	0	1	1	1	0	1	0	0	
Step 15	0	1	1	1	0	1	1	0	—	

Fig. 7.8 (b) Circuit with shift registers for dividing by polynomial $X^5 + X^4 + X^2 + 1$

Part (b) is a table that shows the step-by-step operation as the input is applied one bit at a time. Each row of the table shows the values currently stored in the five shift-register elements. In addition, the row shows the values that appear at the outputs of the three exclusive or circuits. Finally, the row shows the value of the next input bit, which is available for the operation of the next step.

Because no feedback occurs until a 1-dividend bit arrives at the most significant end of the register, the first five operations are simple shifts. Whenever a 1 bit arrives at the left end of the register (C_4), a 1 is subtracted (exclusive or) from the second (C_3), fourth (C_1), and sixth (input) bits on the next shift. This is identical to the binary long-division process illustrated earlier. The process continues through all the bits of the message, plus five zero bits. These latter bits account for shifting M to be left five position to accommodate the FCS. After the last bit is processed, the shift register contains the remainder (FCS), which can then be transmitted.

At the receiver, the same logic is used. As each bit of M arrives, it is inserted into the shift register. If there have been no errors, the shift register should contain the bit pattern for R at the conclusion of M. The transmitted bits or R now begin to arrive, and the effect is to zero out the register so that, at the conclusion of reception, the register contains all 0s.

Figure 7.9 indicates the general architecture of the shift register implementation of a CRC for the polynomial $P(X) = \sum_{i=0}^{n} a_i X^i$ where $a_0 = a_n = 1$ and all other a_i equal either 0 or 1.

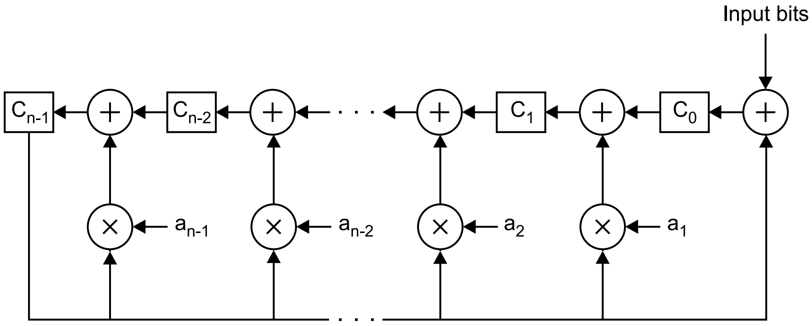


Fig. 7.9 General CRC architecture to implement divisor $1 + a_1X + a_2X^2 + ... + a_{n-1}X^{n-1} + X^n$.

7.5 FLOW CONTROL

Another important design issue that occurs in the data link layer (and higher layers as well) is what to do with a sender that systematically wants to transmit frames faster than the receiver can accept them. This situation can easily occur when the sender is running on a fast (or lightly loaded) computer and the receiver frames out at a high rate until the receiver is completely swamped. Even if the transmission is error free, at a certain point the receiver will simply not be able to handle the frames as they arrive and will start to lose some. Clearly, something has to be done to prevent this situation.

The usual solution is to introduce **flow control** to throttle the sender into sending no faster than the receiver can handle the traffic. This throttling generally requires some kind of a feedback mechanism, so the sender can be made aware of whether or not the receiver is able to keep up.

Various flow control schemes are known, but most of them use the same basic principle. The protocol contains well-defined rules about when a sender may transmit the next frame. These rules often prohibit frames from being sent until the receiver has granted permission, either implicity.

7.6 ELEMENTARY DATA LINK PROTOCOLS

Before dealing the protocols, it is useful to make explicit some of the assumptions underlying the model of communication. To start with, we are assuming that in the physical layer, data link layer, and network layer are independent processes that communicate by passing messages back and forth. In some cases, the physical and data link layer processes will be running on a processor inside a special network I/O chip and the network layer on the main CPU, but other implementations are also possible (*e.g.*, three processes inside a single I/O chip; the physical and data link layers as procedures called by the network layer process, and so on). In any event, treating the three layers as separate processes makes the discussion conceptually cleaner and also serves to emphasize the independence of the layers.

Another key assumption is that machine A wants to sends a long stream of data to machine B using a reliable, connection-oriented service. Later, we will consider the case where B also wants to send data to A simultaneously. A will consider the case where B also wants to send data to A simultaneously. A is assumed to have an infinite supply of data ready to send and never has to wait for data to be produced. When A's data link layer asks for data, the network layer is always able to comply immediately.

As far as the data link layer is concerned, the packet passed across the interface to it from the network layer is pure data, every bit of which is to be delivered to the destination's network layer. The fact that the destination's network layer may interpret part of the packet as a header is of no concern to the data link layer.

When the data link layer accepts a packet, it encapsulates the packet in a frame by adding a data link header and trailer to it. Thus a frame consists of an embedded packet and some control (header) information. The frame is then transmitted to the other data link layer. We will assume that there exist suitable library procedures to-physical layer to send a frame and from-physical-layer to receiver a frame. The transmitting hardware computes and appends the checksum, so that the data link layer software need not worry about it. The polynomial algorithm discussed earlier in this chapter might be used, for example.

Initially, the receiver has nothing to do. It just sits around waiting for something is happen. In the example protocols of this chapter we indicate that the data link layer is waiting for something to happen by the procedure call wait-for-even (and event). This procedure only returns when something has happened. (*e.g.* a frame has arrived). Upon return, the variable event tells what happened. The set of possible events difference for the various protocols to be described and will be defined separately for each protocol. Note that in a more realistic situation, the data link layer will not sit in a tight loop waiting for an event, as we have suggested, but will receive an interrupt, which will cause it to stop whatever it was doing and go handle the incoming frame. Nevertheless, for simplicity we will ignore all the details of parallel activity within the data link layer and assume that it dedicated full time to handling just our channel.

When a frame arrives at the receiver, the hardware computes the checksum. If the checksum is incorrect (*i.e.*, there was a transmission error), the data link layer is so informed. (event = cksum–err), if the inbound frame arrived undamaged, the data link layer is also informed (event = frame–arrival), so it can acquire the frame for inspection using from-physical-layer. As soon as the receiving data link layer has acquired an undamaged frame, it checks the control information in the header, and if everything is all right, the packet portion is passed to the network layer. Under no circumstances is a frame header ever given to a network layer.

There is a good reason why the network must never be given any part of the frame header. To keep the network and data link protocols completely separate. As long as the network layer knows nothing at all about the data link protocol or the frame format, these things can be changed without requiring changes to the network layer's software. Providing a rigid interface between network layer and data link layer greatly simplifies the software design, because communication protocols in different layers can evolve independently.

Figure 7.10 shows some declarations (in C language) common to many of the protocols to be discussed later. Five data structures are defined there: boolean, seq-nr, packet, framekind, and frame.

Boolean: It is an enumerated type and can take on the values true and false.

Seq-nr: A seq-nr is a small integer used to number the frames, so we can tell them apart. These sequence numbers run from 0 up to and including MAX-SEQ, which is defined in each protocol needing it.

Packet: A packet is the unit of information exchanged between the network layer and the data link layer on the same machine, or between network layer peers. In our model it always contain MAX-PKT bytes, but more realistically it would be of variable length.

Frame: A frame is composed of **four** field kind, seq, ack and info, the first three of which contain control information and the last of which may contain actual data to be transferred. The control fields are collectively called frame header.

Kind: The kind field tells whether or not there are any data in the frame, because some of the protocols distinguish frames containing exclusively control information from these containing data as well.

Seq. and Ack.: The seq. and ack. fields are used for sequence numbers and acknowledgement respectively; their use will be described more detail later.

Info: The info field of a data frame contains a single packet; the info field of a control frame is not used. A more realistic implementation would use a variable length info-field, omitting it altogether for control frames.

It is important to realize the relationship between a packet and a frame. The network layer builds a packet by taking a message from the transport layer and adding the network layer header to it. This packet is passed to the data link layer for inclusion in the info field of an outgoing frame. When the frame arrives at the destination, the data link layer extracts the packet from the frame and passes the packet to the network layer. In this manner, the network layer can act as though machines can exchange packets directly.

A number of procedures are also listed in Figure 7.10. These are library routines whose details are implementation-dependent and whose inner working will not concern us further here. The procedure wait-for-event sits in a tight loop waiting for something to happen, as mentioned earlier. The procedures to-network-layer and from-network-layer are used by the data link layer to pass packets to the network layer and accept packets to the network layer respectively. Note that from-physical-layer and to-physical-layer are used for passing frames between the data link and physical layers, whereas the procedures to-network-layer and from-network-layer are used for passing packets between the data link layer and network layer. In other words, to-network-layer and from-network-layer deal with the interface between layers 2 and 3, whereas from-physical-layer and to-physical-layer deal with the interface between layers 1 and 2.

```
//define MAX-PKT 1024                          /*determines packet size in bytes*/
typedef enum{false, true}boolean;              /*boolean type*/
typedef unsigned int seq_nr;                   /*sequence or ack numbers*/
typedef {unsigned char data [MAX_PKT];
} packet;                                      /*packet definition*/
typedef enum {data, ack, nak}frame kind;       /*frame-kind definition*/
typedef struct {                               /*frames are transported in this layer*/
frame-kind kind;                               /*what kind of a frame is it?*/
seq_nr seq;                                    /*sequence number*/
seq_nr ack;                                    /*acknowledgement number
packet info;                                   /*the network layer packet*/
} frame;
/*Wait for an event to happen return its type in event */
Void wait-for-event (event-type *event);
/*Fetch a packet from the network layer for transmission on the channel*/
Void from-network-layer (Packet *P);
/*Deliver information from an inbound frame to the network layer.*/
Void to-network-layer (Packet *r);
/*Pass the frame to the physical layer for transmission*/
Void to-physical-layer (Frame *s);
/*Start the clock running and enable the timeout event.*/
Void start-time (seq_nr k);
/*Stop the clock and disable the timeout event*/
Void stop-timer (seq_nr k)
/*Start an auxiliary timer and enable the ack-timeout event.*/
Void start-ack-timer (void);
/*Stop the auxiliary timer and disable the ack-times event*/
Void stop-ack-timer (void);
/*Allow the network layer to cause a network-layer-read-event.*/
Void enable-network-layer (void);
/*Forebid the network layer from causing a network-layer-event*/
Void disable-network-layer (void);
/*Macro nice is expanded in-line : Infrement k circularly.*/
# define inc (k) if (k < MAX-SEQ k = k + 1; else k = 0
```

Fig. 7.10 Some definitions needed in the protocols to follow

These definitions are located in the file protocol.*h*. In most of the protocols we assume an unreliable channel that losses entire frames upon occasion. To be able to recover from such calamities, the sending data link layer must start an internal timer or clock whenever it sends a frame. If no reply has been received within a certain predetermined time interval, the clock times out and the data link layer receives an interrupt signal. In our protocols this is handled by allowing the procedure wait-for-event to return even = timeout. The procedures start-timer and stop-timer are used to turn to timer on and off, respectively. Timeouts are possible only when the timer is running. It is explicitly permitted to call start-timer while the timer is running. Such a call simply resets the clock to cause the next time-out after a full timer interval has elapsed. (Unless it is reset or turned of in the meanwhile).

The procedures start-ack-timer and stop-ack-timer are used to control and auxiliary timer used to generate acknowledgements under certain conditions.

The procedures enable-network-layer and disable-network-layer are used in the more sophisticated protocols, where we no longer assume that the network layer always has packets to send. When the data link layer enables the network layer, the network layer is then permitted to interrupt when it has a packet to be sent. We indicate this with event = network–layer-ready. When a network layer is disabled, it may not cause such events. By being careful about when it enables and disables it network layer, the data link layer can prevent the network layer from swamping it with packets for which it has no buffer space.

From sequence numbers are always in the range 0 to MAX-SEQ (inclusive), where MAX-SEQ is different for the different protocols. It is frequently necessary to advance a sequence number by 1 circularly (*i.e.*, MAX-SEQ is followed by 0). The macroinc performs this incrementing. It has been defined as a macro because it is used in-line within the critical path. As we will see later in this book, the factor limiting network performance is often protocol processing, so defining simple operations like this as macros does not affect the readability of the code, but does improve performance. Also, since MAX-SEQ will have different values in different protocols, by making it a macro, it becomes possible to include all the protocols in the same binary without conflict. This ability is useful for the simulator.

An Unrestricted Simplex Protocol

We will consider a protocol that is as simple as can be. Data are transmitted in one direction only. Both the transmitting and receiving network layers are always ready. Processing time can be ignored. Infinite buffer space is available. And best of all, the communication channel between the data link layers never damages or lose frames. This thoroughly unrealistic protocol, is shown in Figure 7.11.

```
Typedef enum {frame-arrival} event-type;
#include “Protocol.h”
Void sender 1 (void)
  {
        frames;                          /*buffer for an outbound frame*/
        packed buffer;                   /*buffer for an outbound packet*/
        while (true) {
        while (true) {
        from-network-layer (& buffer)    /*go get something to sent*/
        s. info = buffer;                /*copy it into s for transmission*/
        to-physical-layer (& s);         /*send it on its way*/
```

```
          start from here (or shift right)
      }   /*tomorrow, and tomorrow, and tomorrow
          creeps in this petty pace from day to day
      }   To the last syllable of recorded time ?*
          Macbeth, V, V*/
          void receiver 1 (void)
       {
            frame r;
            event-type event;       /*filled in by wait, but not used her*
            while (true) {
            wait-for-even (/ event);  /*only possibility is frame-arrival*/
            from-physical-layer (& r);
            to-network-layer (& r.info);
            }
       }
```

Fig. 7.11 An Unrestricted simplex protocol

The protocol consists of two distinct procedures, a sender and a receiver. The sender runs in the data link layer of the source machine, and a receiver runs in the data link layer of the destination machine. No sequence number or acknowledgements are used here, so MAX-SEQ is not needed. The only event type possible is from-arrival (*i.e.*, the arrival of an undamaged frame).

The sender is an infinite while loop just pumping data out onto the line as fast as it can. The body of the loop consists of three actions: go fetch a packet from the (always obliging) network layer, construct or outbound frame using the variable *s*, and send the frame on its way only the info field of the frame issued by this protocol, because the other fields have to do with error and flow control and there are no errors or flow control restrictions here.

The receiver is equally simple. Initially, it waits for something to happen, the only possibility being the arrival of an undamaged frame. Eventually, the frame arrives and the procedure wait-for-event returns, with event set to frame-arrival (which is ignored anyway). The call to from-physical-layer removes the newly arrived frame from the hardware buffer and puts it in the variable r. Finally, the data portion is passed on to the network layer and the data link layer settles back to wait for the next frame, effectively suspending itself until the frame arrives.

A Simplex STOP-AND-WAIT Protocol

Now we will drop the most unreligistic restriction used in protocol 1: the ability of the receiving network layer to process incoming data infinitely fast (or equivalently, the presence in the receiving data link layer of an infinite amount of buffer space in which to store all incoming frames while they are waiting their respective turns). The communication channel is still assumed to be error free however, and the data traffic is still simplex.

The main problem we have to deal with here is how to prevent the sender from flooding the receiver with the data faster than the latter is able to process it. In essence, if the receiver requires a time Δt to execute from-physical-layer plus to-network-layer, the sender must transmit at an average rate less than one frame per time Δt. Moreover, if we assume that there is no automatic buffering and queueing done within the receiver's hardware, the sender must never transmit a new frame until the old one has been fetched by from-physical-layer, lest that the new one overwrite the old one.

In certain restricted circumstances (*e.g.*, synchronous transmission and a receiving data link layer fully dedicated to processing the one in line), it might be possible for the sender to simply insert a delay into protocol 1 to slow it down sufficiently to keep from swamping the receiver. However, more usually, each data link layer will have several lines to attend to, and the time interval between a frame arriving and its being processed may vary considerably. If the network designers can calculate the worst-cast behaviour of the receiver, they can program the sender to transmit so slowly that even if every frame suffers the maximum delay there will be no overruns. The trouble with this approach is that it is too conservative. It loads to a bandwidth utilization that is far below the optimum, unless the best and worst cases are almost the same (*i.e.*, the variation in the data link layer's reaction time is small).

A more general solution of this dilemma is to have the receiver provide feedback to the sender. After having passed a packet to its network layer, the receiver sends a little dummy frame back to the sender which, in effect, gives the sender permission to transmit the next frame. After having sent a frame, the sender is required by the protocol to bide its time until the little dummy (*i.e.* acknowledgement) frame arrives.

Protocols in which the sender one frame and then waits from an acknowledgement before proceeding are called stop-and-wait. Figure 7.12 gives on example of a simplex stop and wait protocol.

```
typedef enum {frame_arrival} event_type;
include “protocol.h”
Void sender 2 (void)
fraores;        /*buffer for an outbound frame
packet buffer;       /*buffer for an outbound packet*/
event-type event;  /*frame arrival is the only possibility*/
while (true) {
from-network-layer (& buffer)  /*go get something to send*/
s.info = buffer;      /*copy it into s for transmission*/
to-physical-layer (& s);    /*do not proceed until given the go ahead*/
  }
}
void receiver 2 (void)
{
frame r, s;   /*buffers for frames*/
```

```
event-type event; /*frame-arrival is the only possibility*/
while (true) {
wait-for-event (& event);  /*only possibility is frame-arrival*/
from-physical-layer (& r);  /*go get the inbound frame*/
to-network-layer (& r. info);       /*pass the data to the network layer*/
to-physical-layer (& s);    /*send a dummy frame to awaken sender*/
   }
}
```

Fig. 7.12 A Simplex stop and wait protocol

As in protocol 1, the sender starts out be fetching a packet from the network layer, using it to construct a frame and sending it on its way. Only now, unlike in protocol 1, the sender must wait until an acknowledgement frame arrives before looping back and fetching the next packet from the network layer. The sending data link layer need not even inspect the incoming frame: there is only one possibility.

The only difference between receiver 1 and receiver 2 is that after delivering a packet to the network layer, receiver 2 sends an acknowledgement frame back to the sender before entering the wait loop again. Because only the arrival of the frameback at the sender is important, not its contents, the receivers need not put any particular information in it.

A Simplex Protocol for a Noisy Channel

Consider the normal situation of a communication, etc. that makes error. Frames may be either damaged or loop completely. However, we assume that if a frame is damaged transmit, the receiver hardware will detect this when it computes the checksum. If the frame is damaged in such a way that the checksum is nevertheless correct, an exceedingly. Unlikely occurrence, this protocol (and all other protocols) can fail (*i.e.*, deliver an incorrect packet to the network layer).

At first glance it might seem that a variation of protocol 2 could work: adding a timer the sender could send a frame, but the receiver would only send an acknowledgement frame if the data were correctly received. If a damaged frame arrived at the receiver, it would be discarded. After a while the sender would time out and send the frame again. This process would be repeated until the frame finally arrived intact.

The above scheme has a fatal flow in it. Think about the problem and try to discover what might go wrong before reading further.

To see what might go wrong, remember that it is the task of the data link layer processes to provide error free transparent communication between network layers processes. The network layer on machine, A gives a series of packets delivered to the network layer on machine B, by its data link layer. In particular, the network layer on B has no way of knowing that a packet has been lost or duplicated, so the data link layer must guarantee that no combination of transmission error, no matter how unlikely, can cause a duplicate packet to be delivered to a network layer.

Consider the following scenario:

1. The network layer on A gives packet 1 to its data link layer. The packet is correctly received at B and passed to the network layer on B. B sends an acknowledgement frame back to A.
2. The acknowledgement frame gets lost completely. It just never arrives at all. Life would be a great deal simpler if the channel only mangled and lost data frames and not control frames, but sad to say, the channel is not very discriminating.
3. The data link layer on A eventually times out. Not having received an acknowledgement, it (incorrectly) assumes that its data frame was lost or damaged and sends the frame containing packet 1 again.
4. The duplicate frame also arrives at data link layer on B perfectly and is unwittingly passed to the network layer there. If A is sending a file to B, part of the file will be duplicated (*i.e.*, the copy of the file made by B will be incorrect and the error will not have been detected). In other words, the protocol will fail.

Clearly, what is needed is some way for the receiver to be able to distinguish a frame that it is seeing for the first time from a retransmission. The obvious way to achieve this is to have the sender put a sequence number in the header of each frame it sends. Then the receiver can check the sequence number of each arriving frame to see if it is a new frame or a duplicate to be discarded.

Since a small frame header is desirable, the question arises: What is the minimum number for the sequence number? The only ambiguity in this protocol is between a frame, m and its direct successor, $m + 1$. If frame m is lost or damaged, the receiver will not acknowledge it, so the sender will keep trying to send it. Once it has been correctly received, the receive will send an acknowledgement back to the sender. It is here that the potential trouble drop up. Depending upon whether the acknowledgement frame gets back to the sender correctly or not, the sender may try to send m or $m + 1$.

The event that triggers the sender to start sending $m + 2$ is the arrival of an acknowledgement for $m + 1$. But this implies that m has been correctly receive and furthermore that its acknowledgement has also been correctly received by the sender (otherwise, the sender would not have begun with $m + 1$, let alone $m + 2$). As a consequence, the only ambiguity is between a frame and its immediate predecessor or successor, not between the predecessor and successor themselves.

A 1-bit sequence number (0 or 1) is therefore sufficient. At each instant of time, the receiver expects a particular sequence number next. Any arriving frame containing the wrong sequence number is rejected as a duplicate. When a frame containing the correct sequence number arrives, it is accepted, passed to the network layer, and the expected sequence number is incremented modulo 2 (*i.e.* 0 becomes 1 and 1 becomes 0).

An example of this kind of protocol is shown in Figure 7.13. Protocol in which the sender waits for a positive acknowledgement before advancing to the next data item are of then called PAR (Positive Acknowledgement with Retransmission) or ARQ (Automatic Repeat request). Like protocol 2 this one also transmits data only in one direction. Although it can handle lost frames (by timing out), it requires the timeout interval to be long enough to prevent armature timeouts. If the sender times out too early, while the acknowledgement is still on the way, it will send a duplicate.

```
/*protocol 3 (par) allows unidirectional data flow over a annureliable channel*/
# define MAX-SEQ 1      /*must be 1 for protocol 3*/
typedef enum {frame arrival, cksum-err, timeout} event-type;
# include  "protocol.h"
void sender 3 (void)
{
seq_nr next_frame_to_send;    /*seq number of next outgoing frame*/
frame s;      /*scratch variable*/
packet buffer;      /*buffer for an outbound pack just event-type event;
event_type event;
next_frame_to_sent = 0; /*initialize outbound sequence numbers*/
from_network_layer (buffer)    /*fetch first packet*/
while (true) {
s. info = buffer;    /*construct a frame for transmission*/
s. seq = next_frame_to_send;  /*insert sequence number in frame*/
to_physical_layer (& s)   /*send it on its way*/
start_timer (s. seq); /*if (answer takes too long, timeout*/
wait_for_event (& event); /*frame_arrival, cksum-err, timeout*/
if (event = frame_arrival) {
from_network_layer (& buffer); /*get the next one to send 8?
inc (next_frame_to_send); /*invert next_frame_to_send*/
    }
 }
}
void receiver 3 (void)
 {
 seq_nr frame-expected;
 frame r, s;
 event-type event;
 frame-expected = 0
while (true) {
wait_for_event (& event); /*possibilities : frame_arrival, cksum_err*/
if (event = = frame_arrival) { /*a void frame has arrived*/
from_physical_layer (& r) /*go get the newly arrival frame*/
if (r. seq = = frame_expected) {  /*this is what we have been waiting for*/
to_network_layer (& r.info); /*next time expected the other sequence nr*/
 }
 to_physical_layer (& s); /*none of the fields are used*/
    }
 }
}
```

Fig. 7.13 A positive acknowledgement with retransmission protocol

When the previous acknowledgement finally does arrive, the sender will mistakenly think that the just_sent frame is the one being acknowledged and will not realize that it is potentially another acknowledgement frame somewhere "in the pipe". If the next frame sent is lost completely but the extra acknowledgement arrives correctly, the sender was not attempt to retransmit the lost frame, and the protocol will fail. In later protocols the acknowledgement frame will contain information to prevent just this sort of trouble. For the time being, the acknowledgement frames will just be dummies, and will assume a strict alternation of sends and receive.

Protocols differs from its predecessors in that both send and receiver have a variable whose value is remembered while the data link layer is in wait state. The sender remember the sequence number of the next frame to send in next_frame_to_send; the receiver remembers the sequence number of the next frame expected in frame_expected. Each protocol has a short initialization phase before entering the infinite loop.

After transmitting a frame, the sender starts the time running. It was already running, it will be reset to allow another full timer interval. The time interval must be chosen to allow enough time for the frame to get to the receiver, for the receiver to process it in the worst case, and for the acknowledgement frame to propagate back to the sender. Only when that time interval has elapsed is it safe to assume that either the transmitted frame or its acknowledgement has been lost, and to send a duplicate.

After transmitting a frame and starting the timer, sender waits for something exciting to happen. There are three possibilities.

* An acknowledgement frame arrives undamaged, a damaged acknowledgement frame staggers in, or the timer goes off.
* If a valid acknowledgement comes in, the sender fetches the next packet from its network layer and puts it in the buffer, overwriting the previous packet. It also advances sequence numbers.
* If a damaged frame arrives or no frame at all arrive neither the buffer nor the sequence number are changed so that a duplicate can be sent.

When a valid frame arrives at the receiver, its sequence number is checked to see if it is a duplicate. If not, it is accepted, passed to the network layer, and an acknowledgement generated. Duplicates and damaged frames are not passed to the network layer.

7.7 SLIDING WINDOW PROTOCOLS

This is a flow control protocol. When the channel is not error-free and/or the receiver has limited buffer space, flow of frames from sender to receiver must be controlled to prevent buffer overflow and/or duplication of frames at the receiver. A sliding window class of protocols with go-back-N strategy is found in most standard DLC procedures including HDLC.

The Sliding Window Protocol has as its basis the transmitter and receiver each of which maintain a list of frame sequence numbers. In the case of the transmitter, this list corresponds to the frame that can be transmitted, and in the case of the receiver, the list represents the frames that may be received.

Example: If the receiver can allocate n buffers, then it should be able to receive n frames before sending off an acknowledgement. The acknowledgement returned from the receiver to the transmitter includes the sequence number of the next frame expected. This acknowledgement implicitly announces that the receiver is now able to receive the next n frames, starting with the frame specified in the acknowledgement. For multiple frames, the receiver could delay sending an acknowledgement. For example, if the receiver can receive frames 2, 3 and 4 as shown in Figure 7.14, it could withhold sending an acknowledgement until frame 4 arrives correctly. It would then return the acknowledgement with a sequence number of 5, thereby, informing the transmitter that it has correcty received all frames up to 5.

A data link between a sender A and a receiver B is said to be window flow controlled, if there is an upper bound on the number of frames that can be transmitted by A but not yet been acknowledged by B.

This upper bound (a positive integer) is called the Window size or simply the Window. The number of outstanding *i.e.*, unacknowledged frames at any instant should not exceed the window size. Acknowledgements, as discussed earlier, are either contained is special control frames, or are piggybacked on regular data frames in the reverse direction. The sliding window technique is used along with either go-back-N or selective repeat.

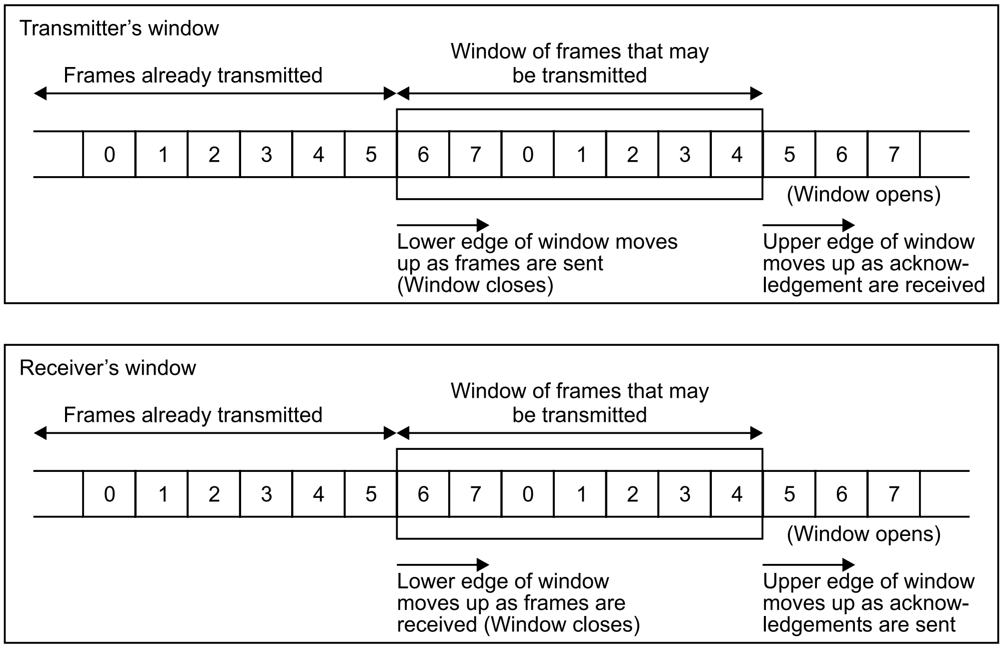

Transmitter's window
Window of frames that may be transmitted
Frames already transmitted
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
(Window opens)
Lower edge of window moves up as frames are sent (Window closes)
Upper edge of window moves up as acknow-ledgement are received
Receiver's window
Window of frames that may be transmitted
Frames already transmitted
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
(Window opens)
Lower edge of window moves up as frames are received (Window closes)
Upper edge of window moves up as acknow-ledgements are sent

Fig. 7.14 Sliding window on the transmitter and receiver

The semantics of an n-bit sliding window protocol is as follows, each transmitted frame contains a sequence number between 0 to (2^n – 1). For stop-and-wait sliding window protocol, n = 1. So that only two sequence numbers, namely 0 and 1 are possible. A common value of n is 3 that allow eight frames to be transmitted without any acknowledgement. The sequence numbers are modulo-2^n. At any point of time, the sender (receiver) maintains a list of consecutive sequence numbers corresponding to frames it is permitted to send receives these frames are said to fall within the "opening" of the sending (receiving) window. It is interesting to note that the receiving and sending window size need not be identical their "opening" range need neither be matched nor be synchronised at every point of time. The open portion of the sending windows represents frames transmitted but as yet not acknowledged. When a new frame appears, it is assigned the next highest sequence number, provided the window is not fully closed, and one advances the upper edge of the closed portion. When an acknowledgement comes in, one making the open portion wider advances the lower edge by one. The closed portion of this window, thus, maintains a list of unacknowledged frames.

On the contrary, the open portion of the receiving window corresponds to the frames that the receiver is waiting to accept. When a frame whose sequence number is equal to the lower edge of the open portion, is received, its acknowledgement is generated, and the window is rotated by one. Unlike sending window that is dynamically opened and closed, the receiving window always maintains its initial size.

Since any frame within the open portion of sending window may have to be retransmitted, the sender must keep all these frames in its buffer. Thus the sending buffer size must equal to the window size for the worst case possible.

The sending window mechanism actually differs from one ARQ strategy to another. We discuss them one by one. In fact, a sliding window protocol can be combined with any one of the ARQ strategies to enforce both flow control and error handing in the data link layer protocol itself.

7.7.1 Stop-And-Wait Sliding Window Protocol

For stop-and-wait protocol, the size is 1 ($n = 1$). The sender transmits a frame and waits for its acknowledgement before sending the next one. To explain the sliding window mechanism, let us consider a slightly blown up version of the protocol as given below.

Example: Data is to be transmitted from City A to City B. The data was CRC coded at City A using the divisor 10101. The data 1100100101011 was received at City B. How would you find whether the received data is the exact one sent from City A?

Solution: Dividing the codeword by 10101, we get

```
10101 ) 1100100101011 ( 111110001
        10101
        -----
        11000
         10101
         -----
         11010
          10101
          -----
           11111
           10101
           -----
            10100
             10101
             -----
              11011
              10101
              -----
               1110   Remainder, hence errors are there.
```

We allow the sequence number to be 2-bit (*i.e.*, 0 through 3). The corresponding window operations are shown in Figure 7.15. Both sending and receiving windows are set at 0. The sending window ready to transmit frame 0 is received the receiving window is rotated by one to be ready to receive the next frame. The receiver issues the acknowledgement before the window is rotated. When the acknowledgement reaches the sender, sending window again becomes open to transmit the next frame.

The number of the acknowledgement reaches the sender, the sending window again becomes open to transmit the number of the last frame received correctly. If this number matches with the number of outstanding frame in the sender's buffer, the frame is taken to be received properly by the receiver, and the sender takes up the next frame for further transmission. If there is a mismatch, the sender retransmits the frame from the buffer. This is a pascal-like procedure for the protocol.

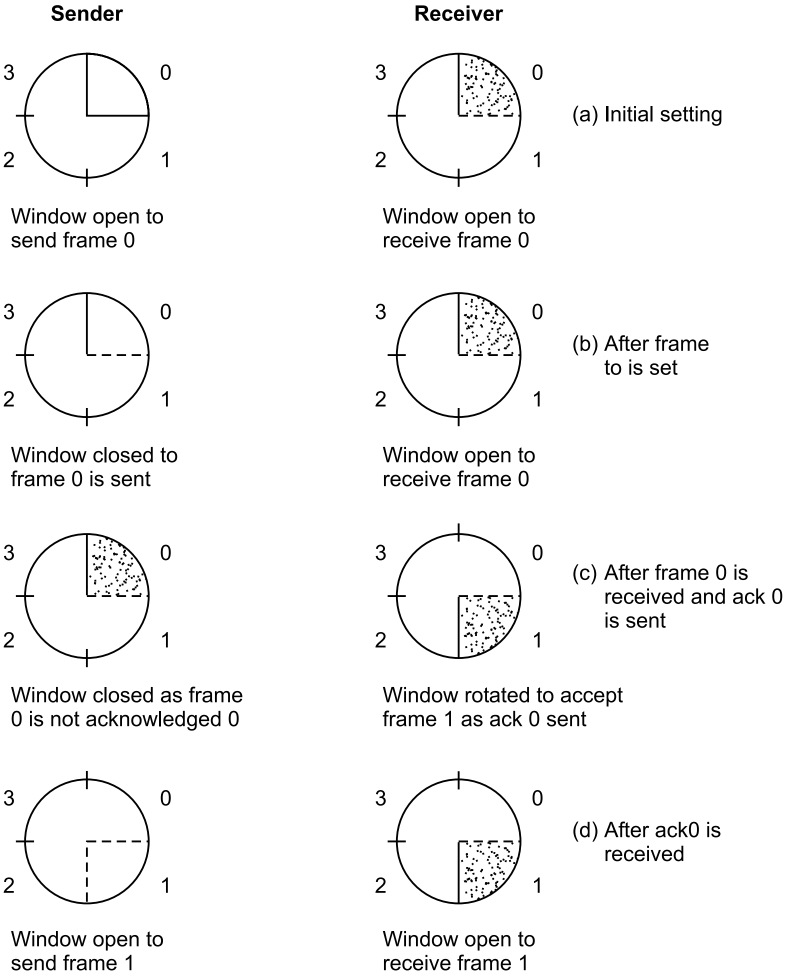

Sender
Receiver
3
0
2
1
(a) Initial setting
Window open to send frame 0
Window open to receive frame 0
(b) After frame to is set
Window closed to frame 0 is sent
Window open to receive frame 0
(c) After frame 0 is received and ack 0 is sent
Window closed as frame 0 is not acknowledged 0
Window rotated to accept frame 1 as ack 0 sent
(d) After ack0 is received
Window open to send frame 1
Window open to receive frame 1

Fig. 7.15 Stop-and-wait sliding window protocol

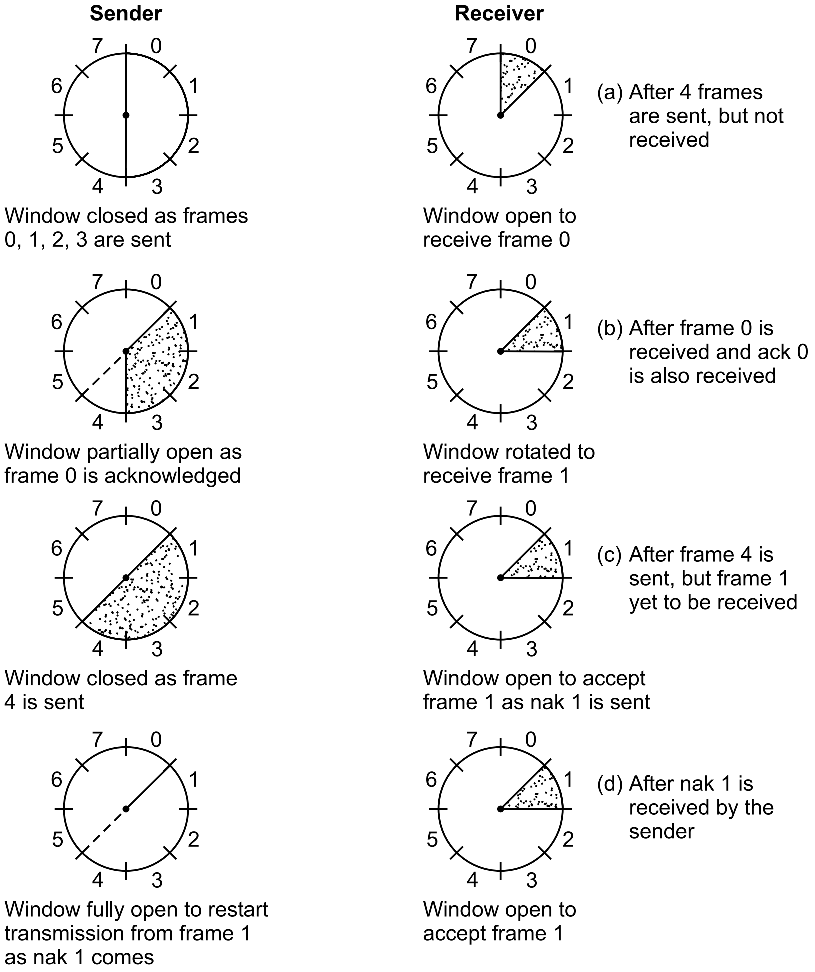
7.7.2 Sliding Window Protocol with Go-Back-N

This is a protocol used in most common DLC procedures. It corresponds to a sending window of any arbitrary size, but a receiving window of size one. Thus, the receiver does not allow any out-of-sequence frame is received. The receiving window rotates only when the desired frame is received without an error. The receiver buffer size can be only one. But, since a transmitter may have to retransmit all the unacknowledged frames in future, the sender buffer size must be at least equal to the sending window size.

The semantics of go-back-N allows all the outstanding frames, upto the sequence number (N – 1), to be automatically acknowledged, if an ACK or NAK comes in for frame N. The sending window is accordingly a multiple frame. This is an important feature considering the fact that some off the ACK frames, bearing sequence numbers less than N, may be lost.

Moreover, the sender maintains a timer for each outstanding frame. If no explicit/implicit ACK comes for a frame before it clock times out, the frame is retransmitted immediately. So the time-out mechanism has to be there to take care of the loss of ACK/NAK frames.

Consider Figure 7.16



Window closed as frames 0, 1, 2, 3 are sent

Window open to receive frame 0

(a) After 4 frames are sent, but not received

Window partially open as frame 0 is acknowledged

Window rotated to receive frame 1

(b) After frame 0 is received and ack 0 is also received

Window closed as frame 4 is sent

Window open to accept frame 1 as nak 1 is sent

(c) After frame 4 is sent, but frame 1 yet to be received

Window fully open to restart transmission from frame 1 as nak 1 comes

Window open to accept frame 1

(d) After nak 1 is received by the sender

Fig. 7.16 Sliding window protocol with Go-back-N

The sender window size is 4, and the receiver window size has to be 1. To clarify the working of the protocol, we assume the sequence numbers between 0 and 7. Initially, the sending window is fully closed after having sent frames 0, 1, 2 and 3. The receiving window is, however, ready to accept only the first frame namely, frame 0.

Let us assume that this frame is receiving without any error, and ACK-0 also reaches the sender correctly so the sending window open partially to allow transmission of frame 4, where as the receiving window rotates by one to be ready to accept frame 1. But frame 1 appears as a damaged one to the receiver who immediately issues 0 NAK frames.

This situation is shown in Figure 7.16 (c). When this NAK 1 arrives at the sender, the sending window reopens from frame 1 because now the sender has to retransmit all the frames following and including frame 1. The receiver, however, remains at the same position.

If the ACK 0 is lost, but NAK 1 arrives before the timeout for frame 1 occurs, what would be the scenario is left as an exercise.

If the receiver buffer space becomes a more serious constraint that the bandwidth available, this protocol is the preferred one.

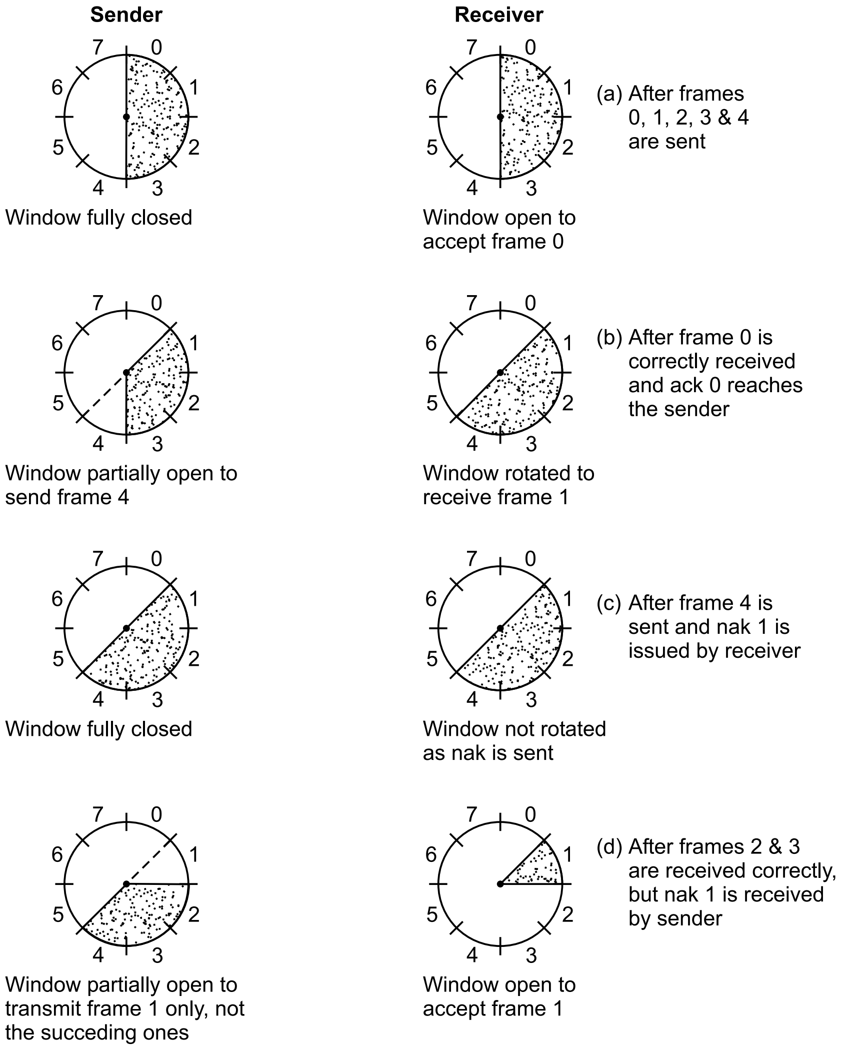
7.7.3 Sliding Window Protocol with Selective Repeat

This protocol is similar to the previous one, except that the receiver window size can be greater than one. Thus, the receiver is allowed to accept frames out of sequence, and it need not discard frames merely because an earlier frame was damaged or lost. The sending window, as usual, closes and opens dynamically with the transmission of frames and reception of positive acknowledgements. A negative acknowledgement forces the retransmission of the only frame, which contains error, not the succeeding ones. This saves precious bandwidth.

The receiving window, on the other hand, always maintains its initial size. When a frame arrives correctly, its sequence number is first checked to find out whether it falls within the open window. If so, it is received irrespective of sequence, but it is not passed to the network layer until all the previous frames have been received correctly and delivered to the network layer. This necessitates a large buffer space at receiver, and the space must be big enough to hold K messages in the worst case, where K is the receiving window size.

If the receiving window size K is kept equal to the range of the sequence number (*i.e.*, $K = 2^n$, where n is the bit length of the sequence number), non-sequential receipt may cause problems. This is due to the fact that, every time the receiving window completes one full rotation, it is identical to the previous window. Hence, the new range of valid sequence numbers overlaps the old one. As a result, if all the previous acknowledgements are lost, presently incoming packets are actually retransmitted duplicates on the earlier frames. But the receiver is completely ignorant of this situation. This is why the receiver window is normally kept bigger than the maximum sequence number. However, to ensure no overlap, K should be exactly half of 2^n, *i.e.*, $K = 2^n - 1$. For instance, if 4 bits are used for sequence numbers, the window size should be eight so only eight unacknowledged frames will be outstanding at any instant.

Hence, both receiving and sending buffer sizes should be at least eight. In the odd halves of the transmission, the receiver window will allow frames 0 to 7 to be accepted, in the even halves the window will rotate to accept frames 8 through 15. This will help to prevent duplication even in the event of acknowledgement failures.



Window fully closed

Window open to accept frame 0

(a) After frames 0, 1, 2, 3 & 4 are sent

Window partially open to send frame 4

Window rotated to receive frame 1

(b) After frame 0 is correctly received and ack 0 reaches the sender

Window fully closed

Window not rotated as nak is sent

(c) After frame 4 is sent and nak 1 is issued by receiver

Window partially open to transmit frame 1 only, not the succeding ones

Window open to accept frame 1

(d) After frames 2 & 3 are received correctly, but nak 1 is received by sender

Fig. 7.17 Sliding window protocol with selective repeat.

Problem: Show the design of a 3 bit sequence number and a maximum window size of seven using sliding window technique.

Solution: We have a 3 bit sequence number and a maximum window size of seven. Initially, both the transmitter and receiver have a window size of seven (in the range [0, 6]). The transmitter sequence number list and window is on the left hand side of the figure 7.17, and the receiver's list and window is on the right hand side of the Figure 7.18.

After transmitting 3 frames (F_0, F_1, F_2) without acknowledgement, the transmitter's window has shrunk to four frames, indicating that it can transmit up to four frames starting with frame number 3. The receiver sends back ACK 3 to the transmitter informing it that it has successfully received all frames upto frame number 2 and is ready to receive frame number 3. In fact, it is able to receive seven frames, starting with frame number 3. With this acknowledgement, the transmitter is now permitted to transmit up to seven frames.

The transmitter then proceeds to send frames F_3, F_4, F_5 and F_6. However, the receiver send back an ACK 4, indicating that it has received frame 3 (F_3) and allowing the transmitter to send frames 4 through to 2. But by the time that this acknowledgement reaches the transmitter, it has already sent off frames 4, 5 and 6. The result is that the transmitter may open its window to permit only four frames to be sent, starting with frame 7.

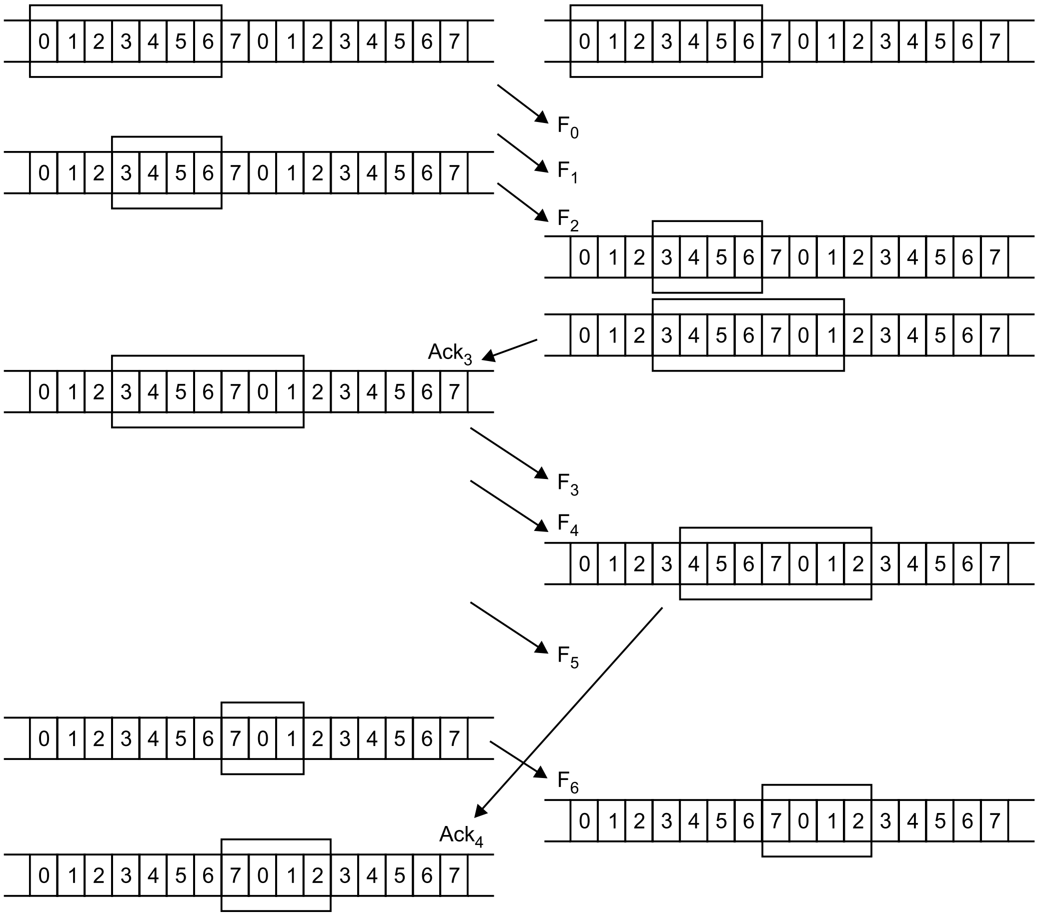


Fig. 7.18 Transmitter and receiver sliding window

7.8 HDLC (HIGH LEVEL DATA LINK CONTROL)

7.8.1 Introduction

IBM began developing DLC protocols way back in 1962 IBM introduced BSC in 1964 and then came up with SDLC. IBM submitted SDLC to ANSI and ISO for acceptance as US and international standards respectively.

The ISO, who first developed COP, went on to modify SDLC to develop a bidirectional data transport mechanism using a single delimiter (flag) and a bit stuffing method of ensuring transparency. This became the HDLC standard in 1976. A SDLC is often referred to as a subnet of HDLC, we will be discussing only about HDLC in this unit.

7.8.2 Types of Stations

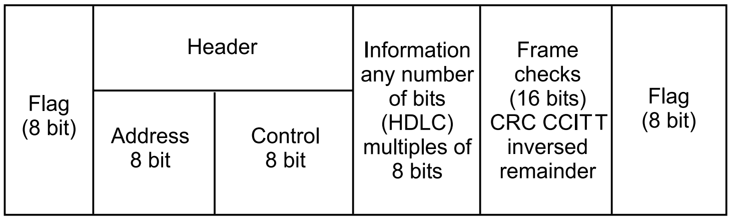
For purposes of traffic regulation, following are the two types of stations in the HDLC data link:

1. Primary Station (commanding)
2. Secondary Station (responding)

The primary station is responsible for

- Data link,
- Retaining control of the link at all times.
- Initiating error recovery procedures.
- Initiating all transmission flow to and from the primary.

All other stations other than the primary are called as secondary stations.



Flag (8 bit)	Header		Information any number of bits (HDLC) multiples of 8 bits	Frame checks (16 bits) CRC CCITT inversed remainder	Flag (8 bit)
	Address 8 bit	Control 8 bit			

Figure 7.19 Framing in HDLC

HDLC is structured to perform in a polling mode, where the primary stations initiates and controls transmission. HDLC does not operate in a full demand mode, where either a secondary or a primary can authorise a transmission. However, individual protocol such as SDLC, have asynchronous response mode whereby a secondary can initiate a transmission when authorised by the central control.

HDLC Logical Link Configuration

HDLC is designed for a variety of physical link types such as multi-access, point-to-point, multipoint and loop. Depending upon which physical link configuration is used, HDLC may have one of the following three logical link configurations:

1. Unbalanced
2. Balanced
3. Symmetrical

1. Unbalanced Link Configurations: An unbalanced link configuration, which may be point-to-point or multipoint, has a primary station and one or more secondary stations as shown in Figure 7.20 and Figure 7.21. The configuration is unbalanced because the primary station is control of each secondary station.

2. Balanced Link Configuration: A balanced link configuration consists of two combined stations with a point-to-point connection.

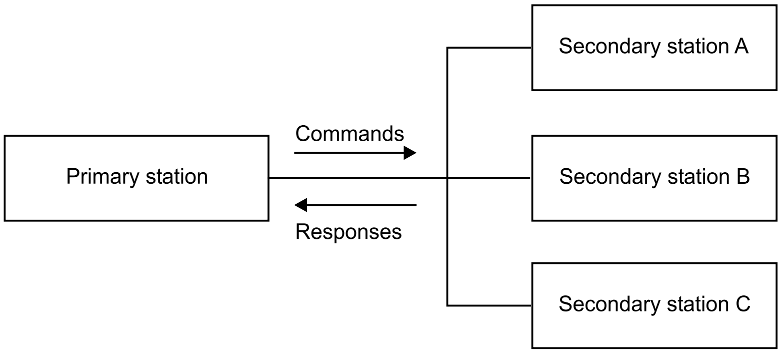


Fig. 7.20 HDLC unbalanced configuration (Multipoint)

Here both stations have identical data transfer and link control capability as shown in Figure 7.21. There is no master slave relationship between the stations. It is supported only in point-to-point operations. X.25 LAPB uses this kind of balanced configuration.

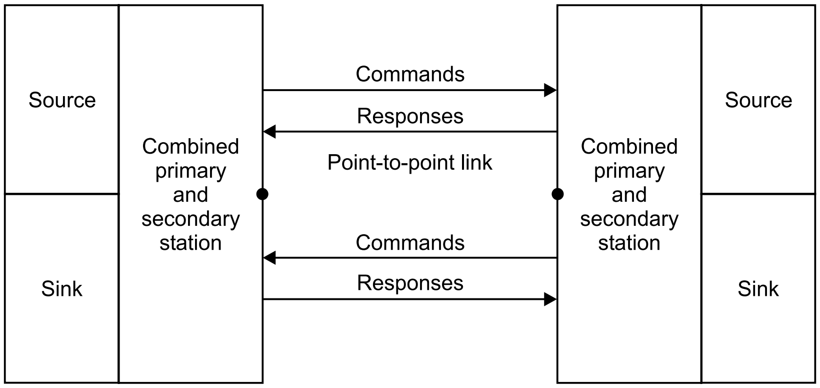

Source
Sink
Combined primary and secondary station
Commands
Responses
Point-to-point link
Commands
Responses
Combined primary and secondary station
Source
Sink

Fig. 7.21 HDLC balanced link configuration

3. Symmetrical Link Configuration: A symmetrical link configuration consists of two unbalanced point-to-point logical station configurations connected together symmetrically and multiplexed on a single physical data link as shown in Figure 7.22.

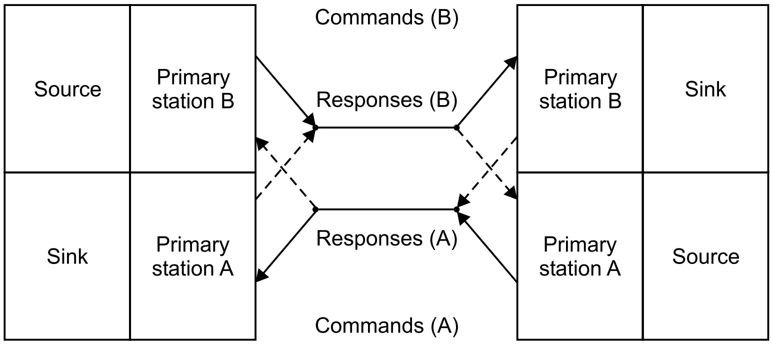


Fig. 7.22 HDLC symmetrical link configuration

This configuration is used in X.25 LAP. Note that two primary-to-secondary logical channels are required over a single physical channel in symmetrical configuration.

In addition to point-to-point and multipoint HDLC unbalanced configuration also operates in Loop structure as shown in Figure 7.23.

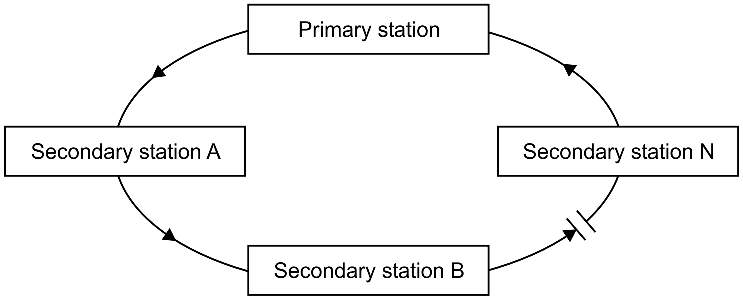


Fig. 7.23 Unbalanced link configuration in loop structure

Here all transmissions flow in the same direction around the loop (transmissions are said to flow down-loop) and pass through the secondary stations. The secondary inspect each transmission. When a secondary finds its address in a frame, the secondary captures that frame for its own use. In unbalanced configuration, a secondary station can transmit only when authorized by primary. Individual secondary stations are polled by the primary, and a response is always required to a specific poll. However, in loop configuration, as each station completes its transmission, the next station down-loop can commence transmission.

HDLC Transmission States

An HDLC data link can be in three of the following states in terms of communications activity:

- **Active:** A data link is active when it is transmitting or receiving.
- **Transient:** When a transmission is about to begin or when a line turnaround is in process, then the line is in transient state.
- **Idle:** In half-duplex mode while turnaround.

The modem has to adjust to the changed line characteristics. Then the link becomes idle.

HDLC Logical States

Communication between any two HDLC stations is conducted in one of the three logical states:

- **ITS (Information Transfer State):** This state is entered after the mode is established. The secondary/combined station transmits and receives information frames.
- **IS (Initialization State):** Communication is under the control of a higher-level system procedure (above HDLC), which will cause secondary/combined station to be initialized by the remote secondary/combined station.

- **LDS (Logically Disconnected State):** Prevents the secondary/combined station from receiving or transmitting information.

7.8.3 Modes of Operation

HDLC Secondary/Combined Station Modes

To begin communication, the logical link is first set up to operate the secondary/ combined stations in one of the three possible modes:

NRM (Normal Response Mode)

Used in unbalanced configuration where secondary initiates transmission of frames only after receiving permission to do so by the primary station.

ARM (Asynchronous Response Mode)

Used in unbalanced configuration, where secondary is permitted to initiate transmission even without explicit permission from primary.

ABM (Asynchronous Balanced Mode)

Used in Balanced Configuration only for point-to-point communications, either of the combined stations can initiate a transmission without receiving permission from the other station.

7.8.4 HDLC Frame Format

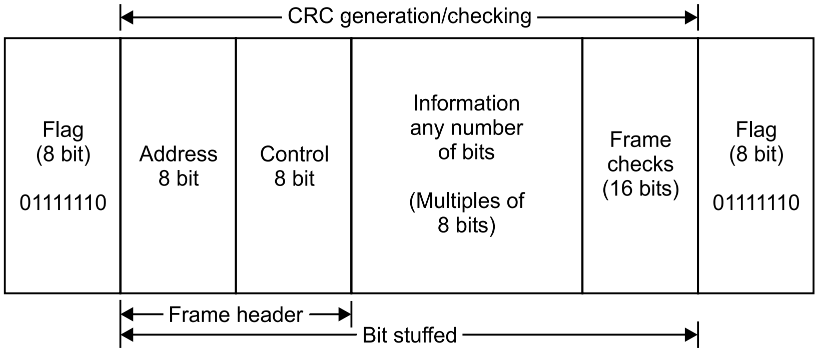


Fig. 7.24 HDLC frame format

All HDLC transmissions in a data line follow the format composition as shown in Figure 7.24 and 7.25 regardless of mode, configuration or procedure. The frame consists of the following fields:

— **Flag: 8** bit opening (01111110) for the start frame.

— **Station Address:** The 8 bit or multiple of 8 bit address identifies the outlaying station that is in communication with the primary.

— **Control Field:** One or two octets of control are used by the primary to control secondary operation, and by the secondary to respond to the primary.

— **Information Field:** Contains the data to be transmitted without constraints on length or bit patterns.

— **Frame Check Sequence:** 16 bit sequence is used (block check) to detect transmission errors.

— **Flag:** 8 bit Closing flag (01111110) for the end of a frame.

8	7	6	5	4	3	2	1
N (R)			P/F	N (S)			0
N (R)			P/F	SS		0	1
N (R)			P/F	MM		1	1

Fig. 7.25 HDLC control field formats

Where MM = Modifier function bit

SS = Supervisory function bit

P/F = Poll bit/Final (Command)

N(S) = Transmitter send sequence number (bit 2 : Lower order bit)

N(R) = Transmitter receive sequence number (bit 6 : Lower order bit)

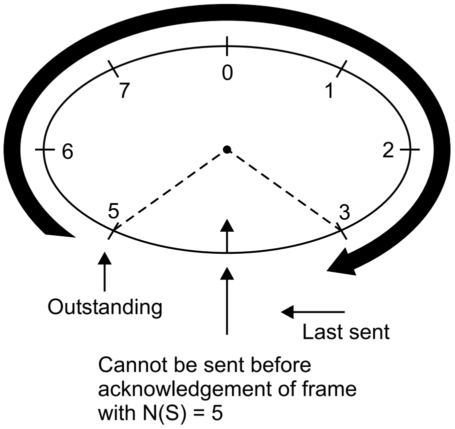

0
1
2
3
5
6
7
Outstanding
Last sent
Cannot be sent before acknowledgement of frame with N(S) = 5

Fig. 7.26 HDLC window operation

HDLC Frame Types

There are three types of HDLC frames, which are as follows:

1. Information Frame
2. Supervisory Frame
3. Unnumbered Frame

1. Information Frame: This frame is used when data is transmitted between primary and secondary.

The control field contains the count of frames sent, N(S), the count of frames received, N(R) and tells if the frame is polling (P) or as final (F) frames in a sequence. N(S) and N(R) are used for error checking and are generated by both primary and secondary stations. A primary station to polls a secondary station, the primary requests a response or a series of responses from secondary. F is used by a secondary station to inform a primary station, when a frame is the last one in a transmission, *i.e.*, the final response to a poll command. If a primary or secondary station does not accepts a final frame, the primary station performs a time out and waits for a response to the poll. When there is no response, the primary station again polls the secondary stations.

2. Supervisory Frame: A frame with a supervisory format in the control field contains no information, and it is used to regulate traffic and request retransmission.

S frames have no information field and are not counted in the N(S) or N(R) frame count sequence. Four commands and responses contained in the supervisory format are:

> Receive Ready (RR)

> Receive Not Ready (RNR)

> Reject (REJ)

> Selective Reject (SREJ)

3. Unnumbered Frame: This frame contain information, but that information is without regard to the send and receive counts.

Also the N(R) and N(S) counts of a station are not changed when a U frame is received. Commands that change modes also reset the frame sequence counts to zero. Of the commands and responses discussed thus far, only P and F are contained in unnumbered format. Some examples of non-sequenced command functions are: Commanding a secondary station in a switched network to go off-hook, exchanging identification between primary and secondary stations and polling secondary stations without changing frame sequence numbers. This frame is used for data link management, activating and initializing secondary stations, reporting of procedural errors not recoverable by retransmission, and controlling response of secondary stations.

7.8.5 Additional Features

HDLC Flow Control

Flow control in HDLC is executed through the well-known window mechanism. The I frames are sequentially numbered and may have a value of 0 through 7 for basic control field format and 127 for the extended control field format. In both instances, the sequence number starts from 0 and cycles through the entire range. Then they re-circulate starting from 0. The maximum number of I frames that's station may have unacknowledged at any given time, is called the window size as shown in Figure 7.26. This number is determined by the storage capability of the sending or receiving station and the link delay. It must be, at least one less than the maximum sequence number.

In the information transfer state, each station maintains a send variable on the I frames it transmits, and receive variable on the I frames it correctly receives. Now each transmitting station has a send variable S that indicates the sequence number of the next I frame to be transmitted. Let us consider an illustration.

If N(S) = 5 has been transmitted, the value of S is incremented to 6. When I frame transmission is aborted S is not incremented. Each receiving station main a receive variable. R equal to the expected N(S) contained in the next I frame received. Upon receipt of an error I frame whose N(S) = 1, R is incremented to (R + 1) mod K, where K is the window size.

HDLC Classes of Procedures

HDLC procedures may be divided into the following classes:

- **Unbalanced Asynchronous:** Used on a multipoint data link with either unbalanced or symmetrical configuration.
- **Unbalanced Normal:** Used on a multipoint data link with either balanced or symmetrical configuration.
- **Balanced Asynchronous:** Used on a balanced point-to-point configuration and is generally more efficient than Unbalanced Asynchronous.

7.9 THE DATA LINK LAYER IN THE INTERNET

The Internet consists of individual machines (hosts and routers), and the communication infrastructure that connects them. Within a single building, LANs are widely used for interconnection, but most of the wise area infrastructure is built up from point-to-point leased lines.

In practice, point-to-point communication is primarily used in two situations.

1. Thousands of organizations have one or more LANS, each with some number of hosts (personal computers, user workstations, servers, and so on) along with a router (or a bridge, which is functionally similar). Often, the routers are interconnected by a backbone LAN. Typically, all connections to the outside world go through one or two routers that have point-to-point leased lines to distant routers. It is these routers and their leased lines that make up the communication subnets on which the Internet is built.
2. The second situation where point-to-point lines play a major role in the Internet is the millions of individuals who have home connections to the Internet using modems and dial-up telephone lines. Usually, what happens is that the user's

home PC calls up an Internet provider, which includes commercial companies like America On-line, Compu-Serve, and the Microsoft Network, but also many universities and companies that provide home Internet connectivity to their students and employee. Sometimes the home PC just functions as a character oriented terminal logged into the Internet service provider's time-sharing system. In this mode, the user can type commands and run programs, but the graphical Internet services, such as the world wide web, are not available. This way of working is called having a shell account.

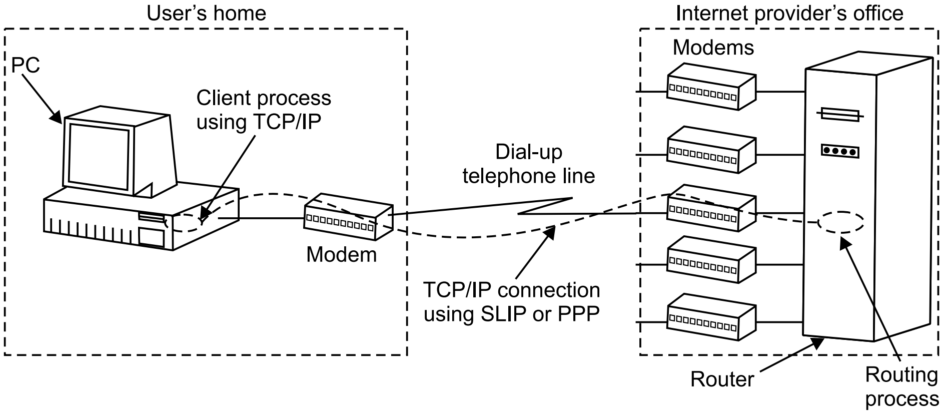

User's home
PC
Client process using TCP/IP
Modem
Dial-up telephone line
TCP/IP connection using SLIP or PPP
Internet provider's office
Modems
Router
Routing process

Fig. 7.27 A Home PC acting as an internet host

Alternatively, the home PC can call an internet service provider's router and then act like a Full-blown Internet host. This method of operation is n different than having a leased line between the PC and the router, except that the connection is terminated when the user ends the session. With this approach, all Internet Services, including the graphical ones, becomes available. A home PC calling an Internet Service provider is illustrate in Figure 7.27.

7.9.1 Slip-Serial Lines IP

SLIP is the older of the two protocols was devised by Rick Adams in 1984 to connect Sun Workstations to the Internet over a dial-up line using a modem. The protocol, which is described in RFC 10 is very simple. The workstation just sends raw IP packets over the line, with a special flag byte (O × CO) at the end for framing. If the flag byte occur inside the IP packet, a form of character stuffing is used, and the two byte sequence (O × DB, O × DC) sent in its place. If O × DB occurs inside the IP packet it, too, is stuffed. Some SLIP implantations attach a flag byte to both the front and back of each IP packet sent.

More recent versions of slip do some TCP and IP header compression. What they do is take advantages of the fact that consecutive packets often have many header fields in common. These are compressed by omitting those field that are the same as the corresponding fields in the previous IP packet. Furthermore, the fields that do differ are not sent in their entirely, but as increments to the previous value, these optimizations are described in RFC 1144.

Although it is still widely used, Slip has some serious problems.

1. It does not do any error detection or correction, so it is up to higher layers to detect and recover from lost, damaged or merged frames.
2. SLIP supports only IP. With the growth of the Internet to compass networks that do not use IP as their native language (*e.g.*, Novell LANs), this restriction is becoming increasingly serious.
3. Each side must know the other's IP address in advance; neither address can be dynamically assigned during setup. Give the current shortage of IP addresses, this limitation is a major issue as it is impossible to give each home Internet user a unique IP address.
4. SLIP does not provide any form of authentication so neither party knows whom it is nearly taking to. With leased lines, this is not an issue, but with dial-up lines it is.
5. SLIP is not an approved Internet standard, so many different (and incompatible) versions exists. This situation does not make interworking easier.

7.9.2 PPP (Point-to-Point Protocol)

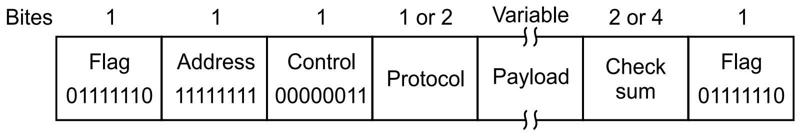
To improve the situation, the LETF set up a group to devise a data link protocol for point-to-point lines that solved all these problems and that could become an official Internet standard. This work culminated in PPP (Point-to-Point Protocol), which is defined in RFC 1661 and further elaborated on in several other RFCs (*e.g.*, RFCs 1662 and 1663). PPP handles error detect supports multiple protocols, allow IP address to be negotiated at connection time, permits authentication, and has many other improvements over SLIP. While many Internet service providers still support both SLIP and PPP, the future clearly lies with PPP, not only for dial-up lines, but also for leased router-router lines.

PPP provides three things:

1. A framing method that unambiguously delineates the end of one frame and the start of the next one. The frame format also handles error detection.
2. A link control protocol for bringing lines up, testing them, negotiating options, and bringing them down again gracefully when they are no longer needed. This protocol is called LCP (Link Control Protocol).
3. A way to negotiate network-layers options in a way that is independent of the network layer protocol to be used. The method chosen is to have a different NCP (Network Control Protocol) for each network layer supported.

The PPP frame format was chosen to closely resemble the HDLC frame format. Since there was no reason to reinvent the wheel. The major difference between PPP and HDLC is that the former is character oriented rather than bit oriented. In particular, PPP like, SLIP, uses character stuffing on dial-up modem lines, so as frames are an integral number of bytes. It is not possible to send a frame consisting of 30.25 bytes, as it is with HDLC. Not only can PPP frames be sent over dial-up telephone lines, but they can also be sent over

SONET or true bit-oriented HDLC lines (*e.g.* for router-router connections). The PPP frames formed is shown in Figure 7.28.



Bites	1	1	1	1 or 2	Variable	2 or 4	1
	Flag 01111110	Address 11111111	Control 00000011	Protocol	Payload	Check sum	Flag 01111110

Fig. 7.28 The PPP full frame format for unnumbered—frame operation

All PPP frames begin with the standard HDLC flag byte (01111110), which is character stuffed if it occurs within the payload field. Next comes the Address field, which is always set to the binary value 11111111 to indicate that all stations are to accept the frame. Using this value avoids the issue of having to assign data link addresses.

The Address field is followed by the Control field, the default value of which is 00000011. This value indicates an unnumbered frame. In other words, PPP does not provide reliable transmission using sequence numbers and acknowledgements as the default. In noisy environments, such as wireless networks, reliable transmission using numbered mode can be used. The exact details are defined in RFC 1663.

Since the Address and control fields are always constant in the default configuration, LCP provides the necessary mechanism for the two parties to negotiate an option to just omit them altogether and save 2 bytes per frame.

The fourth PPP field is the protocol field. Its job is to tell what kind of packet is in the payload field. Codes are defined for LCP, NCP, IP, IPX, AppleTakl, and other protocols. Protocols starting with a 0 bit are network layer protocols such as IP, IPX, OSI OSI CLNP, XNS. Those starting with a 1 bit are used to negotiate other protocol.

These include LCP and a different NCP for each network layer protocol supported. The default size of the protocol field is 2 bytes, but it can be negotiated down to 1 byte using LCP.

The payload field is variable length, up to some negotiated maximum. If the length is not negotiated using LCP during line setup, a default length of 500 bytes is used. Padding may follow the payload if need be.

After the payload field comes the checksum field, which is normally 2 bytes, but a 4 byte checksum can be negotiated.

In summary, PPP is a multiprotocol framing mechanism suitable for use over modems, HDLC bit-serial lines, SONET, and other physical layers. It supports error detection, option negotiation, header compression, and optionally, reliable transmission using HDLC framing.

QUESTIONNARIES

1. Explain the services provided by the data link layer to the network layer.
2. Discuss framing methods.
3. Compare different framing methods wih their advantages and disadvantages.

4. What do you mean by piggy-backing? Explain one bit sliding window protocol.
5. Explain with example, problem in character count method (with diagram).
6. What are the advantages of bit stuffing method over character stuffing?
7. How character are stuffed and destuffed?
8. Why physical layer signal violation schemes are used? What is it's limitations?
9. Discuss different frame formats of HDLC.
10. What are different modes of operation in HDLC? Explain in detail.
11. Explain SLIP and give its disadvantages.
12. Explain go-back-to *n* protocol with neat diagram.
13. What is benefit of selective repeat over go-back-*n*?
14. What is the position of data link layer in the Internet?
15. Explain how CRC detects all single bit errors, multiple bit error, odd number of bits in error and burst error.
16. Explain error detection and correction codes.
17. Explain CRC algorithm generate CRC code. For the data word 110101010 using the divisor 10101.
18. If the average BER is 1 in 10^5, what is probability of having single bit error, single bit correct at least one error in an eight-bit byte?
19. Calculate CRC if generator polynomial is $g(x) = x^3 + x^2 + x + 1$ and $m(x) = 11111$ use EX-OR gates and shift registers.
20. Write short note on:
    - Frame format (PPP)
    - Point to point protocol
    - SLIP
    - Acknowledged connection oriented service
    - Acknowledged connectionless service
    - Unacknowledged connection oriented service

Chapter 8

MEDIUM ACCESS CONTROL METHODS

The protocols used to determine who goes next on a multi-access channel belong to a sublayer of the data link layer called the MAC (Medium Access Control) sublayer. The MAC sublayer is especially important in LANs, nearly all of which use of multi-access channel as the basis of their communication. WANs, in contrast, use point-to-point links, except for satellite networks. Because multi-access channels and LANs are so closely related, in this chapter we will discuss LANs, as well as satellite networks.

8.1 CHANNEL ALLOCATION

8.1.1 Static Channel Allocation

The traditional way of allocating a single channel. Such as a telephone trunk, among multiple competing users is Frequency Division Multiplexing (FDM). If there are N users, the bandwidth is divided into N equal sized portions, echo user being assigned one portion. Since each user has a private frequency band, there is no interference between users. When there is only a small and fixed number of users, each of which has a heavy (buffered) load of traffic (*e.g.*, carries switching offices), FDM is a simple and efficient allocation mechanism.

However, when the number of senders are large in number and continuously varying, or the traffic is bursty, FDM presents some problems. If the spectrum is cut up into N regions, and fewer than N users are currently interested in communicating, a large piece of valuable spectrum will be wasted. If more than N users want to communicate some of them will be denied permission, for lack of bandwidth, even if some of the users who have been assigned a frequency band hardly ever transmit or receive anything.

However, even assuming that the number of users could somehow be held constant at N, dividing the single available channel into static subchannels is inherently inefficient. The basic problem is that when some users are quiescent, their bandwidth is simply lost. They are not using it, and no one else is allowed to use it either. Furthermore, in most computer systems, data traffic is extremely bursty (peak traffic to mean traffic ratios of 1000: 1 are common). Consequently, most of the channels will be idle most of the time.

The poor performance of static FDM can easily be seen from a simple queueing theory calculation. Let us start with the mean time delay, T, for a channel of capacity C bps, with an arrival rate of ν frames/sec, each frame having a length drawn from an exponential probability density function with mean 1/μ bus/frame

$$T = \frac{1}{\mu C - \nu}$$

Now let us divide the single channel up into N independent subchannels, each with capacity C/N bps. The mean input rate on each of the subchannels will now be ν/N. Recomputing T, we get

$$T_{FDM} = \frac{1}{\mu(C/N) - (\nu/N)} = \frac{N}{\mu C - \nu} = NT \qquad ...(1)$$

The mean delay using FDM is N times worse than if all the frames were somehow magically arranged orderly in a big central queue.

8.1.2 Dynamic Channel Allocation

Before we get into the first of the many channel allocation methods to be discussed in this chapter, it is worthwhile carefully formulating the allocation problem. Underlying all the work done in this area are five key assumptions, described below.

1. Station Model

The model consists of N independent stations (computers, telephones, personal communication, etc.), each with a program or user that generates frames for transmission. The probability of a frame being generated in an interval of length Δt is νΔt, where ν is a constant (the arrival rate of new frames). Once a frame has been generated, the station is blocked and does nothing until the frame has been successfully transmitted.

2. Single Channel Assumption

A single channel is available for all communication. All stations can transmit on it and all can receive from it. As far as the hardware is concerned, all stations are equivalent. Although protocol software may assign priorities to them.

3. Collision Assumption

If two frames are transmitted simultaneously, they overlap in time and the resulting signal is garbled. This event is called a collision. All stations can detect collisions. A collided frame must be transmitted again later. There are no errors other than those generated by collisions.

4. Continuous Time

Frame transmission can begin at any instant. There is no master clock dividing time into discrete intervals.

5. Slotted Time

Time is divided into discrete intervals (slots)—frame transmissions always begin at the start of a slot. A slot may contain 0, 1, or more frames, corresponding to an idle slot, a successful transmission, or a collision, respectively.

6. Carrier Sense

Stations can tell if the channel is in use before trying to use it. If the channel is sensed as busy no stations will attempt to use it until it goes idle.

7. No Carrier Sense

Stations can not sense the channel before trying to use it. They just go ahead and transmit. Only later can they determine whether or not the transmission was successful.

Some discussion of these assumptions is in order. The first one says that stations are independent, and that work is generated at a constant rate. It also *implicitly* assumes that each station only has one program or user, so while the station is blocked, no new work is generated. More sophisticated models allow multiprogrammed stations that can generate work, while a station is blocked, but the analysis of these stations is much more complex.

The single channel assumption is also basic, although in some systems (notably spread spectrum), this assumption is relaxed, with surprising results. Also, some LANs, such as token rings, use a mechanism for contention elimination that eliminates collisions.

There are two alternative assumptions about time. Either it is continuous or it is slotted. Some systems use one and some systems use the other. Obviously, for a given system, only one of them holds.

8.2 TYPES OF ACCESS PROTOCOLS

Many algorithms for allocating a multiple access channel are know. In the following sections we will study a representative sample of the more interesting ones and give some examples of their use.

8.2.1 ALOHA

Pure ALOHA

A user station transmits whenever it has traffic to send. Collisions may occur because the separate signals (from several transmitters) propagate on the same channel to the satellite (or another station). The receiver transmits back an ACK or a NAK depending on whether the transmission was successful (the ACK/NAK frequency is typically different from the original transmission channel). A collision necessitates re-transmission, which the transmitter does after a random delay (for continuous stream, like voice or video, there is no re-transmission).

The receiver knows exactly when the transmission occurred.

The overall idea is:

- if you have packet to send, "just do it";
- if packet suffers collision, will try resending later.

Analysing the performance of ALOHA Protocol

The goal for analysis is for the quantitative understanding of performance of the ALOHA Protocol. Let us assume fixed length packets, and packet transmission time is unit time. The throughput (S) is the number of packets successfully (without collision) transmitted per unit time. The offered load (G) is the number of packet transmissions attempted per unit time.

Note : S < G, but S depends on G.

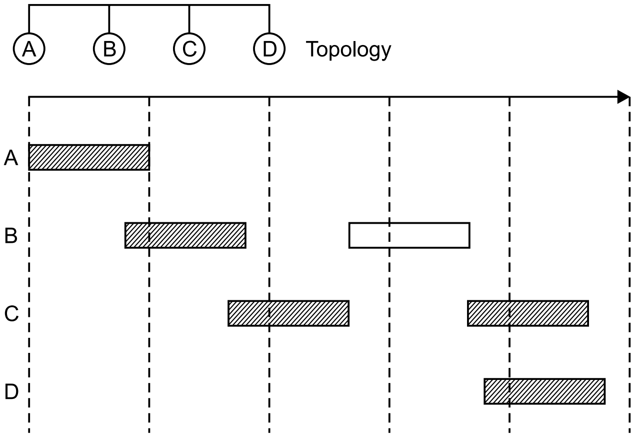


Fig. 8.1 Packet Movement

Let us consider Poisson model *i.e.*, the probability of K packet transmission attempts in *t* time units. In other words we first assume that the probability of P_K transmission attempts per packet time follows a Poisson distribution with a mean G per packet time.

Therefore, $$P_K = \frac{G^K e^{-G}}{K!}$$

The capacity of multiple access protocol is the maximum value of S overall values of G for a given attempted packet transmission.

That is,

$$\begin{aligned} S &= \text{rate attempted packet transmissions} \times \text{Prob. [transmissions successful]} \\ &= G \times \text{Prob. [no other packet's overlap with attempted transmission]} \\ &= G \times \text{Prob. [0 other attempted transmissions in 2 time units]} \\ &= Ge^{-2G} \end{aligned}$$

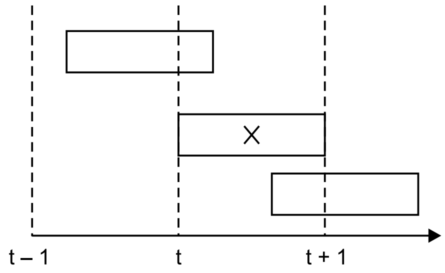


Fig. 8.2 Packet movement with time

ALOHA Throughput

"Erlang" is the unit used to measure throughput, and is defined to be a traffic flow of one packet per transmission time *i.e.*, 1 Erlang is the theoretical maximum throughput lies somewhere in between.

For example, if the best channel utilisation with pure ALOHA is 18%, then we can say that it is 0.18 erlang.

Note that the maximum throughput is 18% of physical channel capacity, *i.e.*, if you buy 1 mb link throughput will never be more than 180 kb!

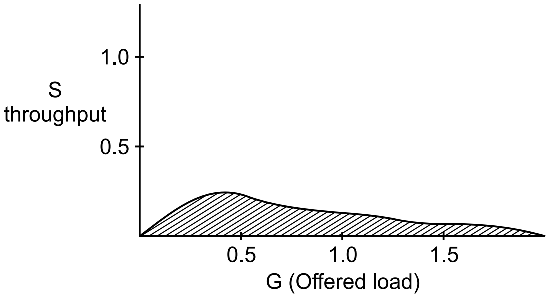


Fig. 8.3 Throughput versus offered load

Slotted ALOHA

Slotted ALOHA requires that common clock be established at the earth stations and the satellite. All three systems (receiver, satellite, and transmitter) agree to transmit the packets in only in *n ms* increments. It follows a synchronous transmission system *i.e.*, the time is divided into slots, the slot size equals fixed packet transmission time and when a packet is ready for transmission, wait until start of next slot. Here the packets overlap completely or not at all as shown in Figure 8.4.

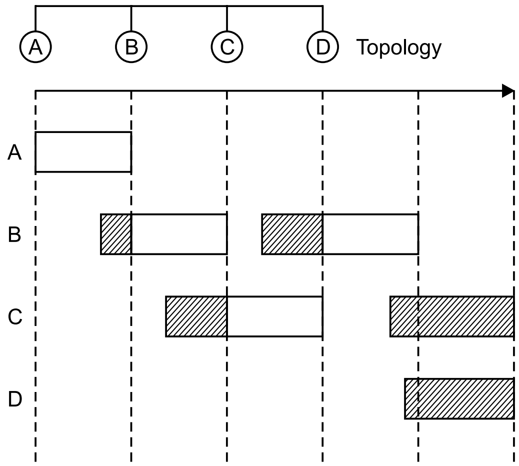


Fig. 8.4 Slotted ALOHA

Performance of Slotted ALOHA

In this case:

$$\begin{aligned} S &= G \times \text{Prob. [no other transmissions overlap]} \\ &= G \times \text{Prob. [0 other attempted transmission]} \\ &= G \times \text{Prob. [0 other arrivals in previous slot]} \\ &= Ge^{-G} \end{aligned}$$

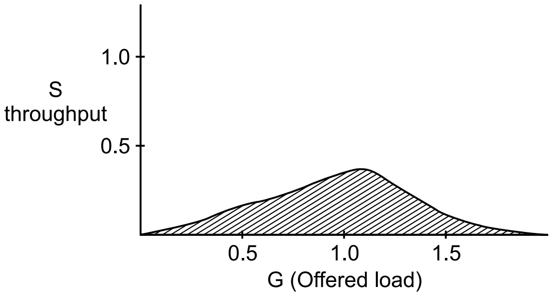


Fig. 8.5 Variations on slotted ALOHA system

There are two schemes of slotted ALOHA referred as reservation ALOHA, or r-ALOHA, which are as given below.

Slotted ALOHA With Non-Owner

The channels slots are combined into an ALOHA frame, the size of which equals or exceeds the up and down propagation delay. For example, if signal takes 100 ms to go to satellite, 100 ms to come down, the frame size is 200 ms. If the slot size is 20 ms, then there are 10 slots in one frame.

In this case—

- A station selects an empty slot in the frame.
- This slot is now reserved for the same user for all successive frames, until he or she runs out of data to transmit (based on our example above, this means the user can transmit for only 20 ms every 200 ms!)
- Once the using station runs out of data to send for reserved slot, it sends an "EOT", following which the slot is empty in the next frame.
- Any other user station can then grab that empty slot, and the process repeats.

Slotted ALOHA With Owner

This system is quite similar to the <<slotted ALOHA with non-owner>> described above, with the following differences:

- Once a station sends an EOT, other station that has data to transmit can use that slot.
- However, if the station that owns the slot returns with data, a collision occurs, and the guest station must relinquish control.
- The owning station must re-transmit its last packet, as must the guest station. The guest station can then use either the slot it owns, or go searching for another vacant slot.

8.2.2 CSMA (Carrier Sensing Multiple Access)

The ALOHA schemes have a potential drawback that before transmitting the transmitter does not even check to see if someone is already transmitting.

CSMA schemes do this, and thus reduce the number of collisions. There are three kinds of CSMA protocols

- 1-Persistent CSMA
- Non-Persistent CSMA
- p-Persistent CSMA

1-Persistent CSMA

This is the simplest form of CSMA protocols. The transmitter just checks the carrier to see that if someone is already transmitting, and if not, then it transmits with a probability of 1. The rest of the protocol behaviour is identical to simple ALOHA. The 1-P is better than ALOHA, but has scope for improvement.

The most important improvement is that the carrier sense is not accurate due to propagation delay *i.e.*, may be the carrier is in use, just that the signal has not reached the transmitter yet. In this case, the transmitter believes that no one else is using the carrier and transmits, resulting in a collision that requires re-transmission.

The other issue is that if two stations A and B check the carrier when C is transmitting, the two continually monitor the channel and transmit as soon as C stops-unfortunately, both of them do so simultaneously resulting in collision. Thus, 1-persistent CSMA protocol waits until the channel is free and then tries to transmit. If there is a collision (which can happen if two stations want to start transmitting at the same time), then they wait for a random time and start over.

Non-Persistent CSMA

In this protocol, the transmitter waits a random amount of time after sensing a busy signal. This modification of 1-Persistent improves on the second problem mentioned in the previous section, *i.e.*, which of the two stations starts transmitting simultaneously as soon as a third stops. As expected, at low traffic intensity, non-persistent CSMA behaves slightly poorly compared to 1-p CSMA; but at normal to high traffic, this method of access can achieve a peak through put of 80%!

Thus, non-persistent CSMA:

```
sense (listen to) channel
if {channel sensed busy}
then wait random time; go to}
else transmit packet.
```

We can say that in case of Non-persistent CSMA rather than sensing the channel continually, the protocol tries again after a random time interval. This approach leads to longer delays but fewer collisions.

p-Persistent CSMA

The *p*-Persistent CSMA is a generalisation of 1-Persistent applied to the slotted channels. The slot length is typically chosen to be the maximum propagation delay. It works as follows.

When a station has data to send it senses the channel. If the channel is idle, the station transmits with probability p. Otherwise it gives another station the opportunity to use the slot with probability (1-p). All stations do this and for all slots, until the frame is transmitted, or the channel is busy. If the channel is already busy, this protocol (unlike the non-persistent variety), continually senses the channel and when it become available, follows the steps given above.

Thus, p-Persistent CSMA:

```
sense (listen to) channel
when {channel sensed idle}
transmit with probability p
else wait random time, go to 1.
```

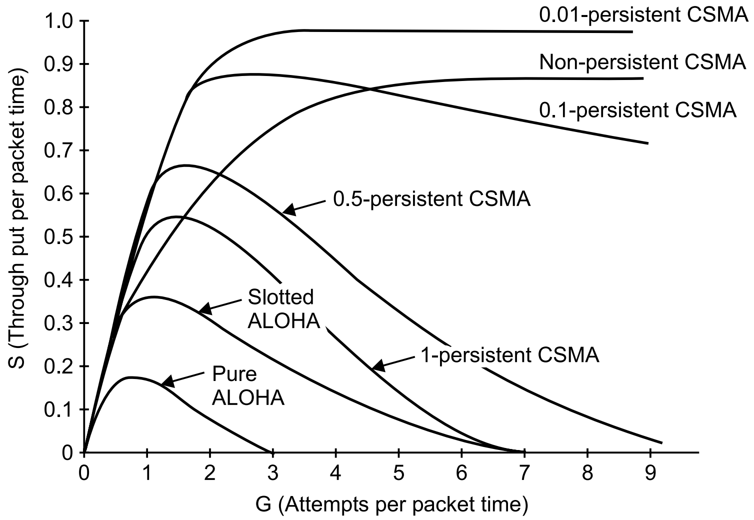


Fig. 8.6 Throughput as a function of the offered traffic and persistence

We can say that in case of p-Persistent CSMA, when a slot is free, a frame is transmitted with probability p. With probability of 1-p, the stations defer to the next slot and so on until the channel is busy or it transmits. When the channel is busy, it acts just like it would with a collision, it waits a random amount of time and starts again.

The given Figure 8.6 illustrates throughput as a function of the offered traffic and persistence for the various random access protocols.

CSMA/CD

After a collision has occurred, it does not make much sense to keep transmitting a frame since it is garbled anyway—particularly in a LAN where collision detection is nearly immediate. Instead in LAN protocols. Like Ethernet, the system will abort transmission after detecting a collision and try again after a random period, assuming another channel has not started. Contention periods last 2t, the distance for a signal to propagate back and forth on the LAN. Special encoding can be used to unambiguously recognise collisions [Two zeros at 0 Volts would not be recognised, but two ones at 5 Volts would produce 10 Volts].

CSMA/CD can be in one of three states: Contention, transmission, or idle. This is shown in Figure 8.7.

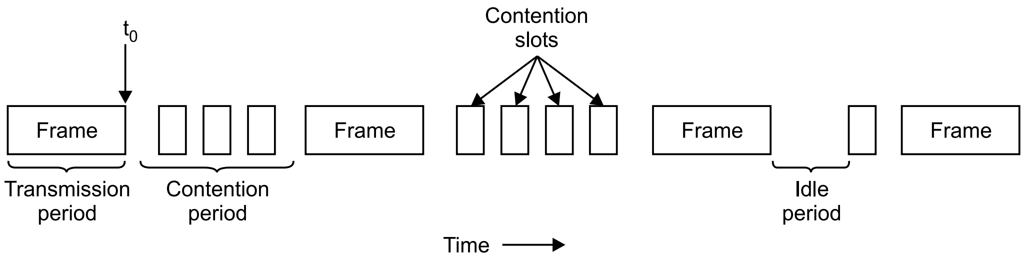


Fig. 8.7 CSMA/CD states

For some networks, any collisions can be a big problem, *e.g.*, optical fiber networks with long propagation distances (large 1) and short frames. Protocols like Bit-map protocol with a designed contention period in which users reserve the next transmission period are one possible solution.

In Bit-map protocol approach, a bit slot is reserved for every member of the network during the contention period. A user that wants to transmit puts *a1* in the slot. The frames are then transmitted in numerical order as shown in Figure 8.8.

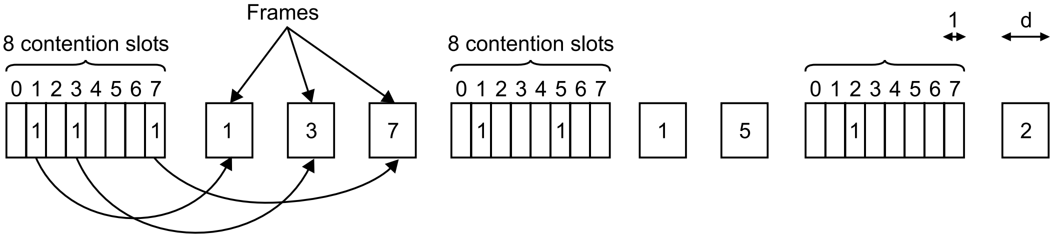


Fig. 8.8 Frames transmitted in numerical order

CSMA/CD however, uses the MAC layer methodology. If channel is idle, station transmits. If the channel is busy, station continues to listen until channel is idle, and then transmits. If a collision is detected in the station, then the sender ceases transmitting current frame and sends out a jamming signal (48 bytes), waits for a small amount of time and then attempts to transmit the message again.

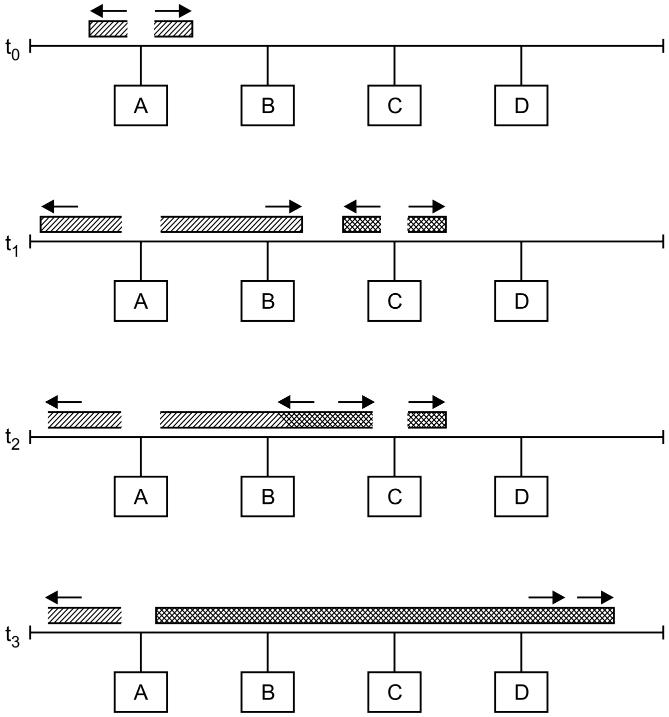


Fig. 8.9 CSMA/CD operation

Figure 8.9 illustrates the technique for the base-band bus. At time to, station A begins transmitting a packet addressed to D. At t_1, both B and C are ready to transmit. B senses a transmission and so defers. C, however, is still unaware of A's transmission and begins its own transmission. When A's transmission reaches C at t_2, C detects a collision and ceases transmission. The effect of the collision propagates back to A, where it is detected after some time t_3, at which A ceases transmission. With CSMA/CD, the amount of wasted capacity is reduced to the time it takes to detect a collision.

8.2.3 Wavelength Division Multiple Access

A different approach to channel allocation is to divide the channel into subchannel using FDM, TDM, or both, and dynamically allocate them as needed. Schemes like this are commonly used on fiber optic LANs in order to permit different conservations to use different wavelengths (*i.e.*, frequencies) at the same time.

A simple way to build an all-optical LAN is to use a passive star coupler. In effect, two fibers from each station are fused to a glass cylinder. One fiber is for output to the cylinder and one is for input from the cylinder. Light output by any station illuminates the cylinder and can be detected by all the other stations. Passive stars can handle hundreds of stations.

To allow multiple transmissions at the same time, the spectrum is divided up into channels (wavelength bands). In this protocol, WDMA (Wavelength Division Multiple Access), each station is assigned two channels. A narrow channel is provided as a control channel to signal the station, and a wide channel is provided to the station can output data frames.

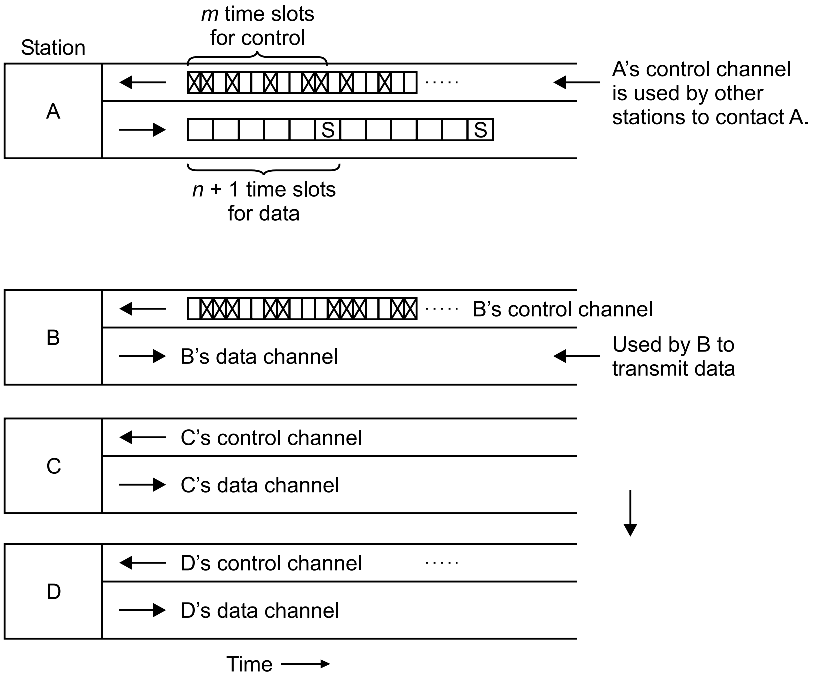


Fig. 8.10 WDMA

Each channel is divided up into groups of time slots as depicted in Figure 8.10: Let us call the number of slots in the control channel *m* and the number of slots in the data channel *n* + 1, where *n* of these are for data and the last one is used by the station to report on its status (mainly, which slots on both channels are free). On both channels, the sequence of slots repeats endlessly, with slot 0 being marked in a special way so latecomers can detect it. All channels are synchronized by a single global clock.

The protocol supports **three classes of traffic:**

1. Constant data rate connection-oriented traffic, such as uncompressed video.
2. Variable data rate connection-oriented traffic, such as file transfer.
3. Datagram traffic, such as UDP packets. For the two connection-oriented protocols, the idea is that for A to communicate with B, it must first insert a CONNECTION REQUEST frame in a free slot on B's control channel. If B accepts, communication can take place on A's data channel.

Each station has **two transmitters** and **two receivers,** as follows:

- A fixed-wavelength receiver for listening to its own control channel.
- A tunable transmitter for sending on their station's control channel.
- A fixed wavelength transmitter for outputting data frames.
- A tunable receiver for selecting a data transmitter to listen to.

In other words, every station listens to its own control channel for incoming requests but has to tune to the transmitter's wavelength to get the data. Wavelength tuning id done by a Fabry-perot or Mach-zehnder interferometer that filters out all wavelengths except the desired wavelength band.

Let us now consider how station A sets up a class 2 communication channel with station B for, say, file transfer. First A tunes its data receiver to B's data channel and waits for the status slot. This slot tells which control slots are currently assigned and which are free. In Figure 8.10, for example, we see that of B's eight control slots, 0, 4, and 5 are free. The rest are occupied (indicated by crosses).

A picks one of the free control slots, say, 4, and inserts it CONNECTION REQUEST message there. Since B constantly monitors its control channel, it sees the request and grants it by assigning slot 4 to A. This assignment is announcement, it knows it has a unidirectional connection. If A asked for a two-way connection, B now repeats the same algorithm with A.

It is possible that at the same time A tried to grab B's control slot 4, C did the same thing. Neither will get it, and will notice the failure by monitoring the status slot in B's control channel. They now each wait a random amount of time and try again later.

At this point, each party has a conflict-free way to send short control message to the other one. To perform the file transfer, A. Now send B a control message saying, for example, "Please watch my next data output slot 3. There is a data frame for you in it." When B gets the control message, it tunes its receiver to A's output channel to read the data frame. Depending on the higher layer protocol, B can use the same mechanism to send back an acknowledgement if it wishes.

Note that a problem arises if both A and C have connections to B and each of them suddenly tells B to look at slots 3. B will pick one of these at random, and the other transmission will be lost.

For constant rate traffic, a variation of this protocol is used. When A askes for a connection, it simultaneously says something like: Is it all right if I send you a frame in every occurrence of slot 3? If B is able to accept (*i.e.*, has no previous commitment for slot 3), a guaranteed bandwidth connection is established. If not, A can try again with a different proposal, depending on which output slots it has free.

Instead of writing a CONNECTION REQUEST message into the control slot it just found (4), it writes a DATA FOR YOU IN SLOT 3 message. If B is free during the next data slot 3, the transmission will succeed. Otherwise, the data frame is lost. In this manner, no connections are ever needed.

Numerous other WDMA protocols have been proposed, differing in the details. Some have one control channel, some have multiple control channels. Some take propagation delay into account, others do not; some make tuning time an explicit part of the model, other ignore it. The protocols also differ in terms of processing complexity, throughput and scalability.

8.3 IEEE STANDARDS

The IEEE has produced several standards for LANs, these standards, collectively known as IEEE 802, include CSMA/CD, token Bus, and token ring. The various standards differ at the physical layer and MAC sub-layer but are compatible at the data link layer. The IEEE 802 standards have been adopted by ANSI as American National Standards, by NIST as government standards, and by ISO as international standards (known as ISO 8802).

The standards are divided into parts, each published as a separate book. The 802.1 standard gives an introduction to the set of standards and defines the interface primitives. The 802.2 standard describes the upper part of the data link layer, which uses the LLC (Logical Link Control) protocol parts 802.3 through 802.5 describe the three LAN standards, the CSMA/CD, token BUS, and token ring standards, respectively. Each standard covers the physical layer and MAC sublayer protocol.

8.3.1 IEEE 802.3: Ethernet

The IEEE 802.3 standard specifies a CSMA/CD bus network that supports 10-Mbps transmission over baseband, broadband, and twisted pair cable. The networking standard closely resembles Ethernet. Both HP and IBM (and others) support the IEEE 802.3 networking standards (HP for their native NS networking product and IBM for their TCP/IP products).

The 802.3 header includes the following:

- **Preamble:** An 8 byte pattern of binary 1s and 0s used to establish synchronisation. The last bit of the preamble is always 0.
- **Start Frame Delimiter:** An 8 bit pattern indicating the formal start of the frame.
- **Destination Address:** An address specifying a specific destination station, a group

of stations, or all stations in the LAN. This address can be 16 bits or 48 bits in length, but all stations in the LAN must adhere to one format or the other.

- **Source Address:** The address of the originating station. This address has the same length requirements as the destination address.
- **Length:** The length measured in bytes, of the actual data, indicating the 802.2 header. This is a 16 bit field.

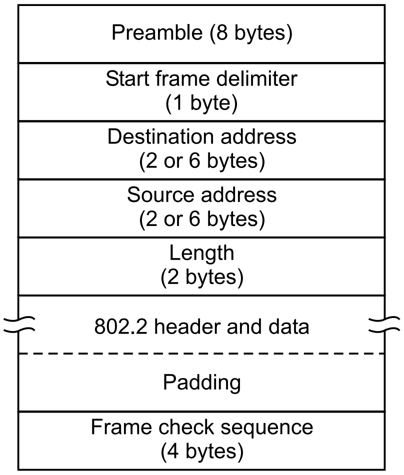

Preamble (8 bytes)
Start frame delimiter (1 byte)
Destination address (2 or 6 bytes)
Source address (2 or 6 bytes)
Length (2 bytes)
802.2 header and data
Padding
Frame check sequence (4 bytes)

Fig. 8.11 IEEE 802.3 CSMA/CD frame

Following the header is the 802.2 header and the actual data. At the end of the data is the 802.3 trailer, which includes:

- **Padding:** Extra, non-data bytes can be inserted into the frame to make the overall frame length more palatable to the physical network.
- **Frame check sequence:** At the end of the frame is a 32 bit Cyclic Redundancy Check (CRC) on the data starting with the destination address the terminating at the end of the data (not including any padding).

Advantages

- Short access delay at low load.
- MAC management is relatively simple, distributed.
- Huge installed base and significant operational experience.

Disadvantages

- Operation at high traffic load is problematic.
- Variable/Unbounded delay-not well suited for real-time applications.
- Substantial analog component (collision detection).

8.3.2 IEEE 802.4 : Token Bus

The IEEE 802.4 specification defines a token passing bus that can operate at speeds of 1, 5, or 10 Mbps. The 802.4 standard is in many ways, a marriage of Ethernet and token ring technologies. The physical topology for 802.4 is a bus, much like in Ethernet, but the MAC-level discipline is a token-passing logical ring (as opposed to a token passing physical ring). Although the 802.4 specification does not have as many active supporters as the 802.3 and 802.5 standards, its popularity is rapidly growing. The format for 802.4 transmission is as follows which is shown in Figure 8.12.

- **Preamble:** One or more bytes used for synchronisation pattern.
- **Start Frame Delimiter:** An 8 bit pattern signalling the start of the Frame.
- **Frame Control:** A 1 byte field used to indicate if the frame. Contains actual data or if it is a control message.

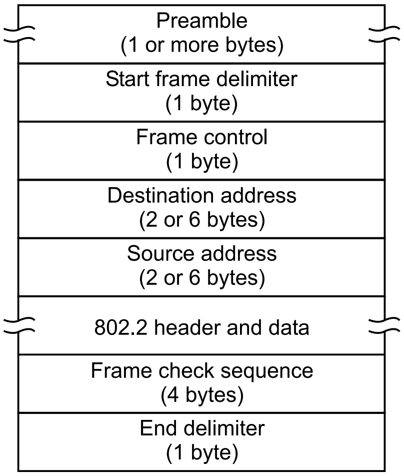


Fig. 8.12 IEEE 802.4 token bus frame

- **Destination Address:** An address specifying a specific destination station, a group of stations, or all stations in the LAN. This address can be 16 bits or 48 bits in length, but all stations in the LAN must adhere to one format or the other.
- **Source Address:** The address of the originating station. This address has the same length requirements as the destination address.

  Following this header is the 802.2 header and the actual data. At the end of the data is the 802.4 trailer, which includes the following:
- ***Frame Check Sequence*:** At the end of the frame is a 32 bit Cyclic Redundancy Check (CRC) on the data starting with the frame control field and terminating at the end of the data.
- ***End Delimiter*:** The 8 bit pattern signaling the end of the frame. The last two bits of this field signal if the frame is the last frame to be transmitted and whether any station has detected an error in the frame.

Advantages

- Access delay at all traffic loads is needed.
- Priority-based access intra-station supports multiple type services.
- Decentralised MAC management is supported.

Disadvantages

- Distributed MAC management requires complicated recovery procedure.
- Substantial analog component.

Frame: The last two bits of this field signal if the frame is the last frame to be transmitted and whether any station has detected an error in the frame.

8.3.3 IEEE 802.5 : Token Ring

The IEEE 802.5 standard specifies a token passing ring operating over shielded twisted pair cables at speeds of 1, 4 or 16 Mpbs. IBM supports this standand in its Token Ring implementation. The 802.5 construction is defined as follows:

- **Start frame delimiter:** An 8 bit-pattern signalling the start of the frame.
- **Access control:** An 8 bit field used for priority and maintenance control. Most important, one bit of this field is the token bit. If set to 1, the frame contains data. If set to 0, the frame is actually a token that can be seized by a station waiting to transmit. Also note that when the token bit is set to 0, the entire frame consists only of the start frame delimiter, the access control byte and the end delimiter, the access control message.
- **Frame control:** A 1 byte field used to indicate if the frame contains actual data or a control message.
- **Destination address:** An address specifying a specific destination station, a group of stations, or all stations in the LAN. This address can be 16 bits or 48 bits in length, but all stations in the LAN must adhere to one format or the other.
- **Source address:** The address of the originating station. This address has the same length requirements as the destination address.

Following this header is the 802.2 header and the actual data. At the end of the data is the 802.5 trailer that includes the following:

- ***Frame Check Sequence*:** At the end of the frame is 32 bit Cyclic Redundancy Check (CRC) on the data starting with the Frame Control field and terminating at the end of data.
- ***End delimiter*:** The 8 bit pattern signalling the end of the frame. The last two bits of this field signal if the frame is the last frame to be transmitted and whether any station has detected an error in the frame.
- ***Frame Status*:** An 8 bit pattern indicating whether a station has recognised the frame and also if the frame has been copied (received).

Advantages

- Access delay at all traffic loads is bounded.
- Media independent
- Star topology eases network management.

Disadvantages

- Priority-based access may starve some stations.
- Centralised monitor station may be a single point of failure.

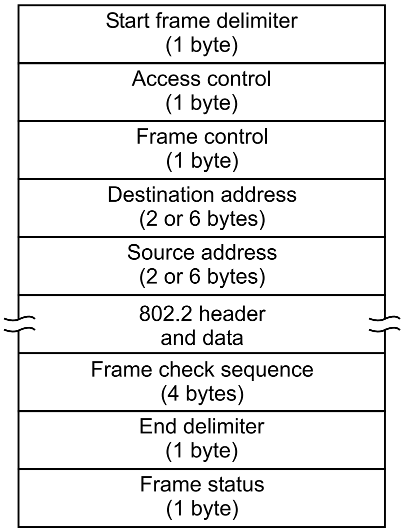


Fig. 8.13 IEEE 802.5 token ring frame

8.4 HIGH SPEED LANS

The 802 LANs and MAN we have just studied are all based an one copper wire (two copper wires for 802.6). For low speeds and short distances, this will do just fine, but for high speeds and longer distances LANs must be based on fiber topics or highly parallel copper networks. Fiber has high bandwidth, is thin and lightweight, is not affected by electromagnetic interference from heavy machinery (important when cabling runs through elevator shafts), power surges, or lightning, and has excellent security because it is nearly impossible to wire tap without detection. Consequently, fast LANs often use fiber. In the following sections we will look at some local area networks that use fiber optics, as well as one extremely high-speed LAN that uses old fashioned copper wire (but lots of it).

8.4.1 FDDI (Fiber Distributed Data Interface)

The Fiber Distributed Data Interface (FDDI) specifies a 100-Mbps token passing, dual-ring LAN using fiber-optic cable. FDDI is frequently used as high-speed backbone technology because of its support for high bandwidth and greater distances than copper. It should be noted that relatively recently, a related copper specification, called Copper Distributed Data

Interface (CDDI) has emerged to provide 100-Mbps service over copper. CDDI is the implementation of FDDI protocols over twisted pair copper wire.

FDDI user dual ring architecture with traffic on each ring flowing in opposite direction (called Counter-rotating). The dual-rings consists of a primary and a secondary ring. During normal operation, the primary ring is used for data transmission, and the secondary ring remains idle. The primary purpose of the dual rings, as will be discussed in detail later in this chapter, is to provide superior reliability and robustness. The Figure 8.14 shows the counter-rotating primary and secondary FDDI rings.

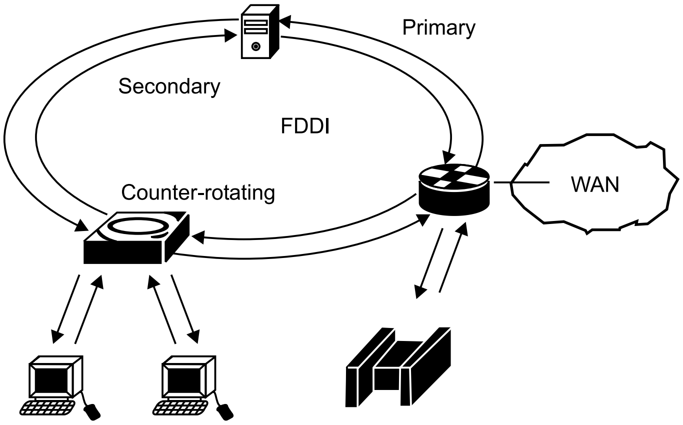


Fig. 8.14 FDDI uses counter rotating primary and secondary rings

Standards

The American National Standards Institute1 (ANSI) X3T9.5 standards committee developed FDDI in the mid-1980s. At the time, high-speed engineering workstations were beginning to tax the bandwidth of existing local area networks (LANs) based on Ethernet and Token Ring. A NEW LAN media was needed that could easily support these workstations and their new distributed applications. At the same time, network reliability had become an increasingly important issue as system managers migrated mission-critical applications from large computers to networks. FDDI was developed to fill these needs. After completing the FDDI specification, ANSI submitted FDDI to the International Organisation for Standardization (ISC), which created an international version of FDDI that is completely compatible with the ANSI standard version.

FDDI Transmission Media

FDDI uses optical fiber as the primary transmission medium, but it also can run over copper cabling. FDDI over copper is referred to as Copper Distributed Data Interface (CDDI). Optical fiber has several advantages over copper media. In particular, security, reliability, and performance all are enhanced with optical fiber media because fiber does not emit electrical signals.

A physical medium that does emit electrical signals (copper) can be tapped and therefore, would permit unauthorised access to the data that is transiting the medium. In addition, fiber is immune to electrical interference from radio frequency interference (RFI) and electromagnetic interference (EMI). Fiber historically has supported much higher bandwidth (throughput potential) than copper, although recent technological advances have made copper capable of transmitting at 100-Mbps. Finally, FDDI allows two kilometres between stations using multi-mode fiber, and even longer distances using a single mode.

FDDI defines two types of optical fiber: Single-mode and multi-mode. A mode is a ray of light that enters the fiber at a particular angle. Multi-mode fiber uses LED as the light-generating devices, while single-mode fiber generally uses lasers.

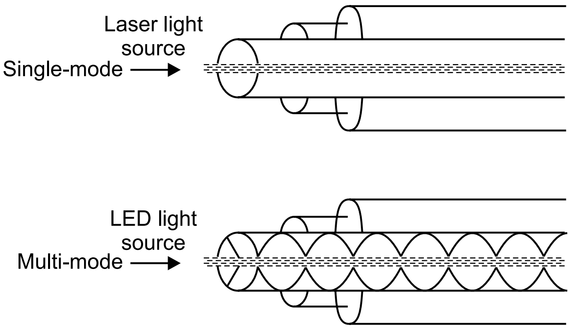


Fig. 8.15 Light sources differ for single/multi-mode fibers

Single-Mode Fiber

These allows only one mode of light to propagate through the fiber. Because only a single-mode of light is used, modal dispersion is not present with single-mode fiber. Therefore, single-mode is capable of delivering considerably higher performance connectivity and over much larger distances. Which is why it generally is used for connectivity between buildings and within environments that are more geographically dispersed. Figure 8.15 depicts single-mode fiber using a laser light source and multi-mode fiber using a light-emitting diode (LED) light source.

Multi-Mode Fiber

These allows multiple mode of light to propagate through the fiber. Because these modes of light enter the fiber at different angles, they will arrive at the end of the fiber at different times. This characteristic is known as modal dispersion. Modal dispersion limits the bandwidth and distances that can be accomplished using multi-mode fibers. For this reason, multi-mode fiber is generally used for connectivity within a building or within a relatively geographically contained environment.

FDDI Frame Format

The FDDI frame format is similar to the format of a Token Ring frame. This is one of the areas where FDDI borrows heavily from earlier LAN technologies, such as Token Ring. FDDI frames can be as large as 4,500 bytes. The following Figure 8.16 shows the frame format of an FDDI data frame and token.

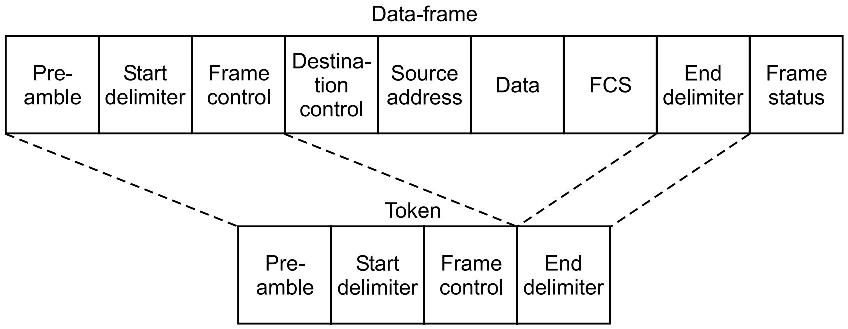


Fig. 8.16 The FDDI frame is similar to that of a token ring frame

The following descriptions summarise the FDDI data frame and token fields illustrated in the Figure 8.16:

- **Preamble:** A unique sequence that prepares each station for an upcoming frame.
- **Start Delimiter:** Indicates the beginning of a frame by employing a signalling pattern that differentiates it from the rest of the frame.
- **Frame Control:** Indicates the size of the address field and whether the frame contains asynchronous or synchronous data, among other control information.
- **Destination Address:** Contains a unicast (singular), multicast (group), or broadcast (every station) address. As with Ethernet and token ring addresses, FDDI destination addresses are 6-bytes long.
- **Source Address:** Identifies the single station that sent the frame. As with Ethernet and Token Ring addresses FDDI source addresses are 6-bytes long.
- **Data:** Contains either information, destined for an upper-layer protocol or control information.
- **Frame Check Sequence (FCS):** Filled by the source station with a calculated Cylic Redundancy Check value dependent on frame contents (as with Token Ring and Ethernet). The destination address recalculates the value to determine whether the frame was damaged in transmit. If so, the frame is discarded.
- **End delimiter:** Contains unique symbols, which cannot be data symbols that indicates the end of the frame.
- **Frame Status:** Allows the source stations to determine whether an error occurred and whether the frame was recognised and copied by a receiving station.

FDDI Specifications

FDDI's specifications are as follows:

- **MAC [Media Access Control]:** It defines how the medium is accessed, including frame format, token handling, addressing, algorithms for calculating redundancy (CRC) value, and error-recovery mechanisms.

- **PHY [Physical Layer Protocol]:** It defines data encoding/decoding procedure clocking requirements, and framing, among other functions.
- **PMD [Physical Medium Dependent]:** It defines the characteristics of the transmission medium, including fiber optic links, power level, bit error rates, optical component and connectors.
- **SMT [Station Management]:** It defines FDDI station configuration, ring configuration, ring control features, including station insertion and removal, initialisation, fault isolation and recovery, scheduling, and statistics collection.

8.4.2 Fast Ethernet

Fast Ethernet refers to a set of specifications developed by the IEEE 802.3 committee to provide a *low cost, Ethernet-compatible, LAN operating at 100-Mbps.* The blank designation for these standards is 100 BASE-T. The committee defined a number of alternatives to be used with different transmission media.

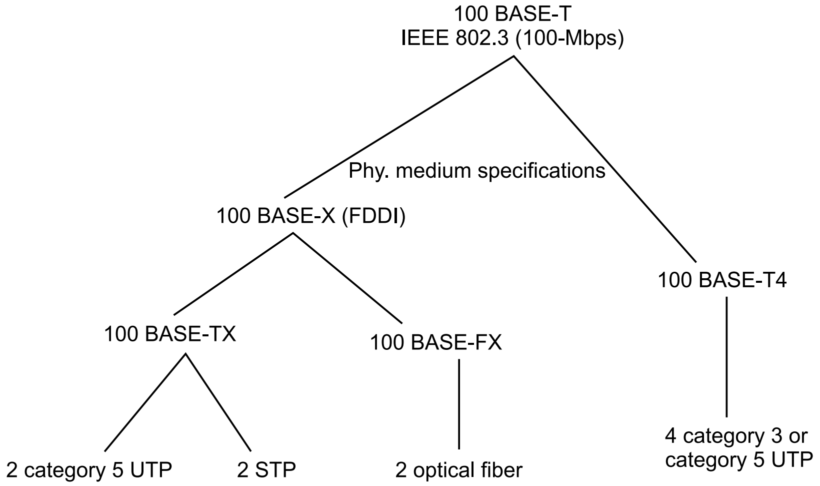


Fig. 8.17 IEEE 802.3 100 BASE-T options

Figure 8.17 shows the terminology used in labelling the specifications and indicates the media used. All of the 100 BASE-T options use the IEEE 802.3 MAC protocol and frame format. 100 BASE-X refers to set of options that use the physical medium specifications originally defined for FDDI covered in the previous section. All of the 100 BASE-X schemes use two physical links between nodes: one for transmission and one for reception. 100 BASE-TX makes use of shielded twisted pair (STP) or high-quality (Category 5) unshielded twisted pair (UTP). 100 BASE-FX uses optical fiber.

In many buildings, each of, the 100 BASE-X options requires the installation of new cable. For such cases, 100 BASE-T4 defines a lower-cost alternative that can use category 3, voice grade UTP in addition to the higher quality category 5 UTP4 to achieve the 100-Mbps data rate over lower-quality cable, 100 BASE-T4 dictates the use of four twisted pair lines between nodes, with the data transmission making use of three pairs in one direction at a time.

For all of the 100 BASE-T options, the topology is similar to that of 10 BASE-T, namely a star-wire topology.

Table 8.1 summarizes key characteristic of the 100 BASE-T options.

Table 8.1 IEEE 802.3 100 BASE-T physical layer medium alternative

	100 BASE-TX		*100 BASE-FX*	*100 BASE-T4*
Transmission Medium	2 Pair, STP,	2 Pair, Category 5 UTP	2 Optical fibers	4 Pair, Category 3, 4 or 5 UTP
Signalling technique	4B5B, NRZI	4B, 5B, NRZI	4B5B, NRZI	8B6T, NRZ
Data rate	100 Mbps	100 Mbps	100 Mbps	100 Mbps
Max. segment length	100 m	100 m	100 m	100 m
Network span	200 m	200 m	400 m	200 m

100 BASE–X

For all at the transmission media specified under 100 BASE-X, a Unidirectional data rate of 100-Mbps is achieved by transmitting over a single link (single twisted pair, single optical fiber). For all of these media, an efficient and effective signal encoding scheme is required. The one chosen was originally defined for FDDI, and can be referred to as 4B/5B-NRZI.

The 100 BASE-X designation includes two physical-medium specifications, one for twisted pair, known as 100 BASE-TX, and one for optical fiber, known as 100 BASE-FX.

100 BASE-TX makes use of two pairs of twisted pair cable, one pair used for transmission and one for reception. Both STP and category 5 UTP are allowed. The MTL-3 signalling scheme is used.

100 BASE-FX makes use of two optical fiber cables, one for transmission and one for reception. With 100 BASE-FX, a means is needed for convert the 4B/5B-NRZI code groups stream into optical signals. The technique used is known as intensity modulation. A binary 1 is represented by a burst or pulse of light, a binary 0 is represented by either the absence of a light pulse or by a light pulse at very low intensity.

100 BASE-T4

100 BASE-T4 is designed to produce a 100 Mbps data rate over lower-quality category 3 cable, thus taking advantage of the large installed base of category 3 cable in office buildings. The specification also indicates that the use of category 5 cable is optional 100 BASE-T4 does not transmit a continuous signal between packets which makes it useful in battery-powered applications.

For 100 BASE-T4 using voice-grade category 3 cable, it is not reasonable to expect to achieve 100-Mbps on a single twisted pair. Instead, 100 BASE-T4 specifies that the data stream to be transmitted is split up into three separate data streams, each with a effective data rate of $33\frac{1}{3}$ M bps. Four twisted pairs are used. Data are transmitted using three pairs and received using three pairs. Thus, two of the pairs must be configured for bidirectional transmission.

As with 100 BASE-X, a simple NRZ encoding scheme is not used for 100 BASE-T4, this would require a signalling rate of 33 Mbps on each twisted pair and does not provide synchronization. Instead, a ternary signalling scheme known as 8B6T is used.

8.5 SATELLITE NETWORKS

Although most multiple access channels are found in LANs, one kind of WAN also uses multiple access channels: Communication satellite based WANs. In the following sections we will briefly study some of the problems that occurs with satellite based wide area networks. We will also look at some of the protocols that have been revised to deal with them.

Communication satellites generally have up to a dozen or so transponders. Each transponder has a beam that covers some portion of the earth below it, ranging from a wide beam 10,000 km across to a spot beam only 250 m across. Stations within the beam area can send frames to the satellite on the uplink frequency. Then satellite then rebroadcasts them on the downlink frequency. Different frequencies are used for uplink and downlink to keep the transponder from going into oscillation satellites that do no on-board processing, but hust echo whatever they hear (most of them), are often called bent pipe satellites.

Each antenna can aim itself at some area, transmit some frames, and then aim at a new area. Aiming is down electronically, but still takes some number of microseconds. The amount of time a beam is pointed to a given area is called a dwell time. For maximum efficiency, it should not be too short or too much, timer will be wasted moving the beam.

Just as with LANs, one of the key designing issues is how to allocate the transponder channels. However, unlike LANs, carrier sensing is impossible due to the 270 msec propagation delay. When a station senses the state of a downlink channel, it has what was going on 270 msec ago. Sensing the uplink channel is generally impossible. As a result, the CSMA/CD protocols cannot be used with satellites. Hence the need for other protocols.

Five classes of protocols are used on the multiple access (uplink) channel: *polling, ALOHA, FDM, TDM,* and *CDMA.* Although we have studied each of these already, satellite operation sometimes adds new twists. The main problem is with the uplink channel, since the downlink channel has only a single sender (the satellite) and thus has no channel allocation problem.

8.5.1 Polling

The traditional way to allocate a single channel among competing users is for somebody to poll them. Having the satellite poll each station in turn to see if it has a frame is prohibitively expensive, given the 270 msec time required for each poll/response sequence.

However, if all the ground stations are also tied to a (typically low bandwidth) packet-switching network, a minor variation of this idea is conceivable. The idea is to arrange all the stations in a logical ring, so each station knows its successor. Around this terrestrial ring circulates a token. The satellite never sees the token. A station is allowed to transmit on the uplink only when it has captured the token. If the number of stations is small and constant, the token transmission time is short, and the busts sent on the uplink channel are much longer than the token rotation time, the scheme is moderately efficient.

8.5.2 ALOHA

Pure ALOHA is easy to implement: every station just sends whenever it wants to. The trouble is that the channel efficiency is only about 18 per cent. Generally, such a low utilization factor is unacceptable for satellites that costs tens of millions of dollars each.

Using slotted ALOHA doubles the efficiency but introduces the problem of how to synchronize all the stations so they all know when each time slot begins. Fortunately, the satellite itself holds the answer, since it is inherently a broadcast medium. On ground station, the *reference station*, periodically transmits a special signal whose rebroadcast is used by all the ground stations as the time origin. If the time slot all have length AT, each station now knows that time slot K begins at a time KΔT after the time origin. Since clocks run at slightly different rates, periodic resynchronization is necessary to keep everyone in phase. An additional complication is that the propagation time from the satellite is different for each ground station, but this effect can be corrected for. To increase the utilization of the uplink channel above 1/e, we could go from the single uplink channel of Figure 8.18 (a), to the duly uplink scheme of Figure 8.18 (b). A station with a frame to transmit chooses one of the two uplink channels at random and sends the frame in the next slot. Each uplink then operates an independent slotted ALOHA channel.

If one of the uplink channels contains a single frame, it is just transmitted in the corresponding downlink slot later. If both channels are successful, the satellite can buffer one of the frames and transmit it during an idle slot later on. Working out the probabilities, it can be shown that given an infinite amount of buffer space, the downlink utilization can go up to 0.736 at a cost of increasing the bandwidth requirements by one half.

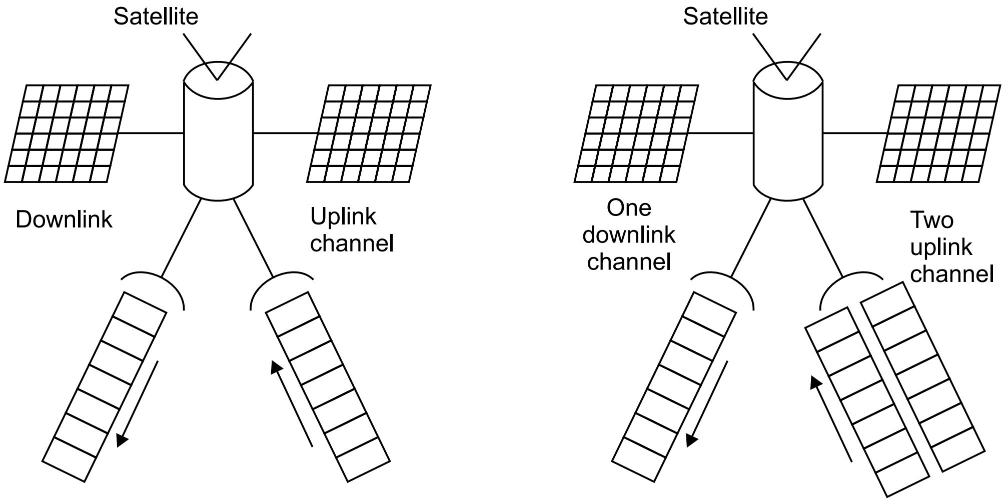


Fig. 8.18 (a) A Standard ALOHA system (b) Adding a second uplink channel

8.5.3 FDM

Frequency division multiplexing is the oldest and probably still the most widely used channel allocation scheme. A typical 36-Mbps transponder might be divided statically into 500 or so 64,000 bps PCM channels, each one operating at its own unique frequency to avoid interfering with the others.

Although simple, FDM also has some drawbacks.

1. Guard bands are needed between the channels to keep the stations separated. This requirement exists because it is not possible to build transmitters that output all their energy in the main band and nothing in the side bands. The amount of bandwidth wasted in the guard bands can be substantial fraction of the total.
2. The stations must be carefully power controlled. If a station puts out too much power in the main band, it will also automatically put out too much power in the side bands, spilling over into adjacent channels and causing interference. Finally, FDM is entirely and analog technique and done not end itself well to implementation in software.

If the number of stations is small and fixed, the frequency channels can be allocated statically in advance. However, if the number of stations, or the load on each one can fluctuate rapidly, some form of dynamic allocation of the frequency bands is needed. One such mechanism is the SPADE system used on some early Intelsat satellites. Each SPADE transponder was divided into 794 simplex (64-Kbps) PCM voice channels, along with a 128-Kbps common signaling channel. The PCM channels were used in pairs to provide full duplex service.

The total transponder bandwidth used was 50-Mbps for the uplink portion and another 50-Mbps for the downlink.

The common signalling channel was divided into units of 50 msec. A unit contained 50 slots of 1 msec (128 bits). Each slot was “owned” by one of (not more than) 50 ground stations. When a ground station had data to send, it picked a currently unused channel at random and wrote the number of that channel in this next 128-bit slot. If the selected channel was still unused when the request was seen on the downlink, the channel was considered allocated and all other stations refrained from trying to acquire it, if two or more stations tried to allocate the same channel in the same frame, a collision occurred and they had to try again later. When a station was finished using its channel, it sent a deallocation message in its slot on the common channel.

8.5.4 TDM

Like FDM, TDM is well understood and widely used in practice. It requires time synchronization for the slots, but this can be provided by a reference station as described for slotted ALOHA above. Similarly to FDM, for a small and unvarying number of stations, the slot assignment can be setup in advance and never changed but for a varying number of stations, or a fixed number of stations with time-varying loads, time slots must be assigned dynamically.

Slot assignment can be done in a centralized or a decentralized way. As an example of centralized slot assignment. Let us consider the experimental ACTS (Advanced Communication Technology Satellite), which was designed for a few dozen stations (Palmer and White, 1990). ACTS was launched in 1992 and has four independent 110-Mbps TDM channels, two uplink and two downlink. Each channel is organized as a sequence of 1 msec frames each frame containing 1728 time slot. Each time slot has a 64-bit payload, allowing each one to hold a 64-Kbps voice channel.

The beam can be switched from one geographical area to another, but since moving the beam takes several slot times, channels originating or terminating in the same geographic area are normally assigned to contiguous time slots to increase dwell time and minimize knowledge of station geography to minimize the number of wasted time slots. For this and other reasons, time slot management is done by one of the ground stations, the MCS (Master Control Station).

The basic operations of ACTs is a continuous three-step process, each step taking 1 msec.

* Step 1: The satellite receives a frame and store it in a 1728 entry onboard RAM.
* Step 2: An onboard computer copies each input entry to the corresponding output entry, (for antenna).
* Step 3: The output frame is transmitted on the downlink.

Initially, each station is assigned at least one time slot. To acquire additional channels (for new voice calls), a station sends a short request message to the MCS. Similarly, it can release an existing channel with a message to the MCS. These messages make use of a small number of overhead bits and provide a special control channel to the MCS with a capacity of about 13 messages/sec per station. The channels are dedicated, there is no contention for them.

Dynamic TDM slot allocation is also possible. Below we will discuss three schemes. In each of these, TDM frames are divided into time slots, with each slot having a (temporary) owner. Only the owner may use a time slot.

The first scheme assumes that there are more slots than stations, so each station can be assigned a home slot (Binder, 1975). If there are more slots than stations, the extra slots are not assigned to anyone. If the owner of a slot does not want it during the current group, it goes idle. An empty slot is a signal to everyone else that the owner has no traffic. During the next frame, the slot becomes available to anyone who wants it, on a contention (ALOHA) basis.

If the owner wants to retrieve “his” home slot, be transmits a frame, thus forcing a collision (if there was other traffic). After a collision, everyone except the owner must desist from using the slot in the next frame. Thus the owner can always begin transmitting within two frame times in the Worst case. At low channel utilization the system does not perform as well as normal slotted ALOHA. Since after each collision, the collides must abstain for one frame to see if the owner wants the slot back. Figure 8.19 (a) shows a frame with eight slot, seven of which are owned by G, A, F, E, B, C and D, respectively. The eight slot is not anyone and can be fought over.

A second scheme is applicable even when the number of stations is unknown and varying. In this method, slots do not have permanent owners, as in Binder's. Instead, stations compete for slots using slotted ALOHA. Whenever a transmission on is successful the station making the successful transmission is entitled to that slot in the next frame as well. Thus, as long as a station has data to send, it can continue doing so indefinitely. In essence the proposal allows a dynamic mix of slotted ALOHA and TDM, with the number of slots devoted to each. Varying with deman Figure 8.19 (b) also shows a frame with eight slots. Initially, E is using the last slot, but after two frames, it no longer needs it. It lies idle for one frame, and then D picks it up and keeps it until it is done

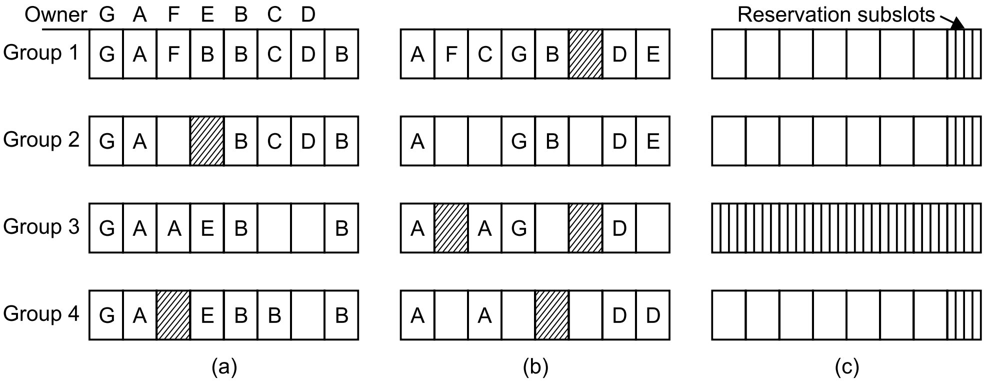


Fig. 8.19 Reservation schemes (a) Binder (b) Crowther (c) Roberts

A third scheme, requires stations to make advance requests before transmitting. Each frame contains, say, one special slot [the last one in Figure 8.19 (c)] which is divided into V smaller subslots used to make reservations. When a station wants to send data it broadcasts a short request frame in randomly-chosen reservation subslot. If the reservation is successful (*i.e.*, no collision) then the next regular slot (or slots) is reserved. At all times everyone must keep track of the queue length (number of slots reserved), so that when any station makes a successful reservation it will know how many data slots to skip before transmitting. Stations need not keep track of who is queued up, they merely need to know how long the queue is. When the queue length drops to zero, all slots revert to reservation subslots, to speed up the reservation process.

8.5.5 CDMA

The CDMA avoids the times synchronization problem and also the channel allocation problem. It is completely decentralized and fully dynamic.

CDMA has *three main disadvantages:*

1. The capacity of a CDMA channel in the presence of noise and uncoordinated stations is typically lower then what TDM can achieve.
2. With 128 chips/bit (a common vote), although the bit rate may not be high, the chip rate will be, necessitating a fast (read: expensive) transmitter.

3. Few practicing engineers actually understand CDMA, which generally does not increase the chances of their using it, even if it is the best method for a particular application.

Nevertheless, CDMA has been used by the military for decades and is now becoming more common in commercial applications as well.

QUESTIONARIES

1. Explain static channel allocation in LAN's and MAN's with possible problem.
2. Explain key assumptions on which dynamic channel allocation is based.
3. Explain ALOHA scheme? What is slotted ALOHA?
4. Compare collision based protocols with collision free protocols. Explain bit-map protocol.
5. Explain a binary count down protocol. How the efficiency of this protocol can be increased to 100%.
6. Explain WDMA with a neat diagram.
7. Explain 802.3 standard.
8. What are different types of cabling for 802.3.
9. Explain MAC sublayer and protocol of 802.3.
10. Explain frame format of 802.3.
11. Explain token bus MAC sublayer protocol.
12. Explain logical ring maintenance of token bus.
13. Draw and explain token ring.
14. Write short note on:
    - 1-persistance
    - p-persistance
    - Non-persistance
    - FDDI
    - Fast Ethernet
    - Satellite networks

Chapter 9

NETWORK LAYER

INTRODUCTION

The network layer is concerned with getting packets from the source all the way to the destination may requires making many hops at intermediate routers along the way. This function clearly contrasts with that of the data link layer, which has the more modest goal of just moving frames from one end of a wire to the other. Thus the network layer is the lowest layer that deals with end-to-end transmission.

When two many packets are present in the subnet, performance degrades. This situation is called congestion, here we discussing about all the concepts of congestion control.

9.1 DESIGN ISSUES

In the following section we will provide an introduction to some of the issues that the designers of the network layer must grapple with.

Services provided to the Transport Layer

The network layer provides services to the transport layer at the network layer/transport interface. This interface is often especially important for another reason. It frequently is the interface between the carrier and the customer, that is, the boundary of the subnet. The carrier often has control of the protocols and interfaces up to and including the network layer. Its job is to deliver packets given to it by its customers. For this reasons, this interface must be especially well defined.

The network layer services have been designed with the following goals in mind:

1. The services should be independent of the subnet technology.
2. The transport layer should be shielded from the number, type and topology of the subnets present.

3. The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs and WANs.

Given these goals, the designers of the network layer. This freedom often degenerates into a ranging battle between two warring factions. The discussion centres on the question of whether the network layer should provide connection-oriented service or connectionless service.

One camp (represented by the Internet Community) argues that the subnet's job is moving bits around and nothing else. In their view (based on nearly 30 years of actual experience with a real, working computer network), the subnet is inherently unreliable, no matter how it is designed. Therefore, the hosts should accept the fact that it is unreliable and do error control (*i.e.*, error detection and correction) and flow control themselves.

This viewpoint leads quickly to the conclusion that the network service should be connectionless, with primitives SEND PACKET and RECEIVE PACKET, and little else. In particular, no packet ordering and flow control should be done, because the hosts are going to do that anyway, and there is probably little to be gained by doing it twice. Furthermore, each packet must carry the full destination address, because each packet sent is carried independently of its predecessors, if any.

The other camp (represented by the telephone companies) argues that the subnet should provide a (reasonably) reliable, connection-oriented service. In this view, connections should have the following properties:

1. Before sending data, a network layer process on the sending side must set up a connection to its peer on the receiving side. This connection, which is given a special identifier, is then used until all the data have been sent, at which time it is explicitly released.
2. When a connection is set up, the two processes can enter into a negotiation about the parameters, quality and cost of the service to be provided.
3. Communication is in both directions, and packets are delivered in sequence.
4. Flow control is provided automatically to prevent a fast sender from dumping packets into the pipe at a higher rate than the receiver can take them out, thus leading to overflow.

Other properties, such as guaranteed deliver, explicit confirmation of delivery, and high priority packets are optional. A connectionless service is like the postal system, and connection-oriented service is like the telephone system.

The argument between connection-oriented and connectionless service really has to do with where to put the complexity. In the connection-oriented service, it is in the network layer (subnet), in the Connectionless Service, it is in the transport layer (hosts). Supports of connectionless service say that user computing power has become cheap, so that there is no reason not to put the complexity in the hosts. Furthermore, they argue that the subnet is a major (inter) national investment that will last for decades, so it should not be cluttered up with features that may become obsolete quickly but will have to be calculated into the price structure of many years. Furthermore, some applications, such as digitized voice and real-time data collection may regard speedy delivery as much more important than accurate delivery.

On the other hand, supporters of connection-oriented service say that most users are not interested in running complex transport layer protocols in their machines what they want is reliable, trouble-free connections. Furthermore, some services, such as real time audio and video are much easier to provide on top of a connection-oriented network layer than on top of a connectionless network layer.

Although it is rarely discussed in these terms, two separate issues are involved here.

1. Whether the network is connection-oriented (setup required) or connectionless (no setup required).
2. Whether it is reliable (no lost, duplicated, or garbled packets) or unreliable (packets can be lost, duplicated, or garbled).

In theory, all four combinations exist, but the dominant combinations are reliable connection-oriented and unreliable connectionless, so the other two tend to get lost in the noise.

Table 9.1 Running TCP/IP over an ATM subnet

E-mail	FTP	....
	TCP	
	IP	
	ATM	
	Datalink	
	Physical	

These two camps are represented by our two tunning examples. The Internet has a connectionless network layer, and ATM networks have a connection-oriented network layer. An obvious question arises about how the Internet works when it runs over an ATM-based, Carrier-provided subnet. The answer is that the source host first establishes an ATM network layer connection to the destination host and then sends independent (IP) packets over it, as shown in Table 9.1. Although this approach works, it is inefficient because certain functionality is in both layers. For examples, the ATM network layer guarantees that packets are always delivered in order, but the TCP Code still contains the full mechanism for managing and reordering out-of-order packets.

Internal Organization of the Network Layer

Having looked at the two classes of service the network layer can provide to its users, it is time to see how it works inside. There are basically two different philosophises for organizing the subnet, one using connections and the other working connectionless. In the context of the internal operation of the subnet, a connection is usually called a virtual circuit, in analogy with the physical circuits set up by the telephone system. The independent packets of the connectionless organization are called datagrams, in analogy with telegrams.

Virtual circuits are generally used in subnets whose primary service is connection-oriented, so we will describe them in that context. The idea behind virtual circuits is to avoid having to choose a new route for every packet or cell sent. Instead, when a connection is established, a route from the source machine to the destination machine is chosen as part of the connection setup and remembered. That route is used for all traffic flowing over the connection, exactly the same way that the telephone system works. When the connection is released, the virtual circuit is also terminated.

In contrast, with a datagram subnet no routes are worked out, in advance, if the service is connection-oriented. Each packet sent is routed independently of its predecessors. Successive packets may follow different routes. While datagram subnets have to do more work, they are also generally more robust and adopt to failures and congestion more easily than virtual circuit subnets. We will discuss the *pros* and *cons* of the two approaches later.

If packets flowing over a given virtual circuit always take the same route through the subnet, each router must remember where to forward packets for each of the currently open virtual circuits passing through it. Every router must maintain a table with one entry per open virtual circuit passing through it. Each packet traveling through the subnet must contain a virtual circuit number field in its header, in addition to sequence numbers, checksums. When a packet carriers at a router, the router knows on which line it arrived and what the virtual circuit number is. Based on only this information the packet must be forwarded on the correct output line.

When a network connection is set up, a virtual circuit number not already in use on that machine is chosen as the connection identifier. Since each machine chooses virtual circuit numbers independently, these numbers have only local significance. If they were globally significant over the whole network, it is likely that two virtual circuits bearing the same global virtual circuit number might pass through some intermediate router, leading to ambiguities.

Because virtual circuits can be initiated from either end, a problem occurs when call setups are propagating in both directions at once along a chain of routers. At some point they have arrived at adjacent routers. Each router must now pick a virtual circuit number to use for the (full-duplex) circuit it is trying to establish. If they have been programmed to choose the lowest number not already in use on the link, they will pick the same number, causing two unrelated virtual circuit over the same physical line to have the same number. When a data packet arrives later, the receiving router has no way of telling whether it is a forward packet on the circuit or a reverse packet on the other. If circuits are simplex, there is no ambiguity.

Note that every process must be required to indicate when it is through using a virtual circuit, so that the virtual circuit can be purged from the router tables to recover the space. In public networks, the motivation is the stick rather than the carrot : users are invariably charged for connect time as well as for data transported. In addition, some provision must be made for dealing with machines that terminate their virtual circuits by crashing rather than politely releasing them when done.

Comparison of Virtual Circuit and Datagram Subnets

Both virtual circuits and datagrams have their supporters and their detractors. We will not attempt to summarize the arguments both ways. The major issues are listed in Table 9.2, although purists could probably find a counter example for everything in the figure.

Inside the subnet, several trade-offs exist between virtual circuits and datagrams. On trade-off is between router memory space and bandwidth. Virtual circuits allow packets to contain circuit numbers instead of full destination addresses. If the packets tend to be fairly short, a full destination addresses in every packet may represent a significant amount of overhead, and hence wasted bandwidth. The price paid for using virtual circuits internally is the table space within the routers. Depending upon the relative cost of communication circuits versus router memory one or the other may be cheaper.

Another trade-off is setup time versus address parsing time. Using virtual circuits requires a setup phase, which takes time and consumes resources. However, figuring out what to do with a data packet in a virtual circuit subnet is easy : the router just uses the circuit number to index into a table to find out where the packets goes. In a datagram subnet, a more complicated procedure is required to determine where the packet goes.

Table 9.2 Comparison of datagram and virtual circuit subnets

Issue	*Datagram subnet*	*VC subnet*
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Subnet does not hold state information	Each VC requires subnet table space
Routing	Each packet is routed independently	Route chosen when VC is setup, all packets follow this route
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Congestion Control	Difficult	Easy if enough buffers can be allocated in advance for each VC

Virtual circuits have some advantages in avoiding congestion within the subnet because resources can be reserved in advance, when the connection is established. Once the packets start arriving, the necessary bandwidth and router capacity will be there. With a datagram subnet, congestion avoidance is more difficult.

For transaction processing systems (*e.g.*, stores calling up to verify credit card purchases), the overhead required to setup and clear a virtual circuit may easily dwarf the use of the circuit. If the majority of the traffic is expected to be of this kind, the use of switched virtual circuits inside the subnet makes little sense. On the other hand, permanent virtual circuits; which are setup manually and last for months or years, may be useful here.

Virtual circuits also have a vulnerability problem. If a router crashes and loses its memory, even if it comes back up a second later, all the virtual circuits passing through it will have to be aborted. In contrast, if a datagram router goes down, only those users whose packets were queued up in the router at the time will suffer, and may be not even all those, depending upon whether they have already been acknowledged or not. The loss of a communication line is fatal to virtual circuits using it but can be easily compensated for it datagrams are used. Datagrams also allow the routers to balance the traffic throughout the subnet, since routes can be changed halfway through a connection.

It is worth explicitly pointing out that the service offered (connection-oriented or connectionless) is a separate issue from the subnet structure (virtual circuit of datagram). In theory, all four combinations are possible. Obviously, a virtual circuit implementation of a connection-oriented service and a datagram implementation of a connectionless service are reasonable. Implementing connections using datagrams also makes sense when the subnet is trying to provide a highly robust service.

The fourth possibility, a connectionless service on top of a virtual circuit subnet, seems strange but certainly occurs. The obvious example is running IP over an ATM subnet. Here it is desired to run an existing connectionless protocol over a new connection-oriented network layer. As mentioned earlier, this is more of an adhoc solution to a problem than a good design. In a new system designed to run over an ATM subnet, one would not normally put a connection less protocol like IP over a connection-oriented network layer like ATM and then layer a connection-oriented transport protocol on top of the connectionless protocol. Examples of all four cases are shown in Table 9.3.

Table 9.3 Examples of different combinations of service and subnet structure

Upper layer	*Type of subnet*	
	Datagram	Virtual circuit
Connectionless	UDP Over IP	UDP Over IP Over ATM
Connection-oriented	TCP Over IP	ATM AAL1 Over ATM

9.2 ROUTING ALGORITHMS

The main function of the network layer is routing packets from the source machine to the destination machine. In most subnets, packets will require multiple hops to make the journey. The only notable exception is for broadcast networks, but every here routing is an issue if the source and destination are not on the same network. The algorithms that choose the routes and the data structures that they are a major area of network layer design.

Routing is the process of finding a path from a source to every destination in the network. The *routing algorithm* is that part of the network layer software responsible for deciding which output line an incoming packet should be transmitted on. If the subnet uses datagrams internally, this decision must be made a new for every arriving data packet since the best route may have changed since last time. If the subnet uses virtual circuits internally, routing decisions are made only when a new virtual circuit is being set up. Thereafter, data packets just follow the previously established route. The latter case is sometimes called *session routing*, because a route remains in force for an entire *user session*.

Regardless of whether routes are chosen independently for each packet or only when new connections are established, there are certain properties that are desirable in a routing algorithm: Correctness and simplicity hardly require comment, but the need for robustness may be less obvious at first. Once a major network comes on the air, it may be expected to run continuously for years without systemwise failures. During that period there will be hardware and software failures of all will change many times. The routing algorithm should be able to cope with changes in the topology and traffic without requiring all jobs in all hosts to be broadest and the network to be rebooted every time some router crashes.

Routing algorithms can be grouped into two major classes : *non-adaptive* and *adaptive.*

Adaptive algorithms

Adaptive algorithms, in contrast, change their routing decisions to reflect changes in the topology, and usually the traffic as well. Adaptive algorithms differ in where they get their information (*e.g.*, locally, from adjacent routers, or from all routers), when they change the routes (*e.g.*, every ΔT sec, when the load changes, or when the topology changes), and what metric is used for optimization (*e.g*., distance, number of hops, or estimated transmit time). Adaptive algorithms are of the following:

1. Distance vector routing
2. Link state routing
3. Broadcast routing

Non-adaptive algorithms

They do not base their routing decisions on measurements or estimates of the current traffic and topology. Instead, the choice of the route to use to get from I to J (for all I and J) is computed in advance, off-line and downloaded to the route when the network is booted. This procedure is sometimes called static routing. Non-adaptive algorithms are of the following:

1. Shortest path routing
2. Flooding
3. Flow-based routing

9.2.1 Shortest Path Routing

Many Practical routing algorithms are based on the notion of a shortest path between two nodes. Here, each communication link is assigned a positive number called it length. A link can have a different length in each direction. Each path between two nodes has a length equal to the sum of the lengths of its links. A shortest path routing algorithm routes each packet along a minimum length (or shortest) path between the origin and destination nodes of the packet. The simplest possibility is for each link to have unit length, in which case a shortest path is simply a path with minimum number of links. More generally, the length of a link may depend on hits transmission capacity and its projected traffic load. The idea here is that a shortest path should contain relatively few and non-congested links, and therefore be desirable for routing.

A more sophisticated alternative is to allow the length of each link to change over time and to depend on the prevailing congestion level of the link. Then a shortest path may adapt to temporary overloads and route packets around points of congestion. This idea is simple but also contains some hidden pitfalls, because by making link lengths dependent on congestion, we introduce a feedback effect between the routing algorithm and the traffic pattern within the network.

An important distributed algorithm for calculating shortest paths to a given destination, known as the Bellman-ford method, has the form

$$\mathrm{D}_i = \mathrm{m}_{ini}\ [\mathrm{d}_{ij} + \mathrm{D}_j]$$

where

D_i → is the estimated shortest distance of node to the destination.

d_{ij} → is the length of the link (i, j).

Each node executes periodically this interaction with the minimum taken over all its neighbours j. Thus $d_{ij} + \mathrm{D}_j$ may be viewed as the estimate of shortest distance from node i to the destination subject to the constant of going through j, and $\min_j (d_{ij} + \mathrm{D}_j)$ may be viewed as the estimate of shortest distance from i to the destination going through the best neighbour.

Shortest path routing has the following two drawbacks:

- It uses only one path per origin-destination Pair thereby potentially limiting the throughut of the network.
- Its capability to adapt to changing traffic conditions is limited by its susceptibility to oscillations, this is due to the abrupt traffic shifts resulting when some of the shortest paths change due to changes in link lengths.

Optimal Routing

It is based on the optimisation of average delay like measure of performance, can eliminate both of these disadvantages by splitting any origin-destination pair traffic at strategic points, and by shifting traffic gradually between alternative paths.

9.2.2 Flooding

During the operation of a data network, it is often necessary to broadcast some information that is, to send this information from an origin node to all other nodes. For example, when there are changes in the network topology due to link failure and repairs, these changes must be transmitted to the entire network.

Broadcasting could also be used as a primitive form of routing packet from a single transmitter to a single receiver. More generally, to a subnet of receivers, this use is generally rather inefficient, but may be sensible because it is simple or because the locations of the receivers within the network are unknown.

A widely used broadcasting method is known as flooding. It operates as follows:

The origin node sends its information in the form of a packet to its neighbours (the nodes to which it is directly connected with a link). In turn the neighbours relay it to their neighbours, and so on, until the packet reaches all nodes in the network. Two additional rules are also observed, which limit the number of packet transmissions.

1. A node will not relay the packet back to the node from which the packet was obtained.
2. A node will transmit the packet to its neighbours at most once, including on the packet the ID number of the origin node a sequence number, which is incremented with each new packet issued by the origin node, can ensure this by storing the highest sequence numbers that are less than or equal of its incident links. Note that with these rules, links need not preserve the order of packet transmissions, the sequence numbers can be used to recognise the correct order.

Example. Consider a flooding that the total number of packet transmissions per packet broadcast lies between L and 2L, where L is the number of bi-directional links of the network. We note also that one can implement a flooding—like algorithm without using sequence numbers.

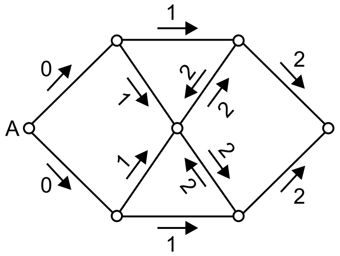


Fig. 9.1 Packet broadcast from A to all other nodes by using flooding

In Figure 9.1, arrows indicate packet transmissions at the times shown. Each packet transmission time is assumed to be one unit.

9.2.3 Flow Based Routing

The algorithms studied so far take only the topology into account. They do not consider the load. Here we will study a static algorithm that uses both topology and load for routing. It is called flow-based routing.

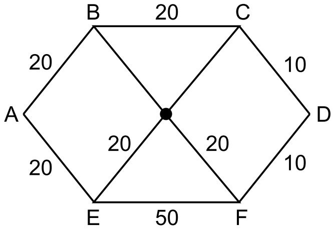
In some networks, the mean data flow between each pair of nodes is relatively stable and predictable. For example, in a corporate network for a retail store chain, each store might send orders, sales reports, inventory updates, and other well defined types of messages to known sites in a predefined pattern, so that the total volume of traffic varies little from day to day. Under conditions in which the average traffic from *i* to *j* is known in advance and, to a reasonable approximation, constant in time, it is possible to analyze the flows mathematically to optimize the routing.

The basic idea behind the analysis is that for a given line, if the capacity and average flow and known, it is possible to compute the mean packet delay on that line from queuing theory. From the mean delays on all the lines, it is straight forward to calculate a flow-weighted average to get the mean packet delay for the whole subnet. The routing problem their reduces to finding the routhing algorithm that produces the minimum average delay for the subnet.

To use this technique, certain-information must be known in advance.

1. The subnet topology must be known.
2. The traffic matrix, f_{ij}, must be given.
3. The line capacity matrix, C_{ij} specifying capacity of each line in bps must be available.
4. A routing algorithm must be chosen.

As an example of this method, consider the full-duplex subnet of Figure 9.2(*a*). The weights on the arcs give the capacities, C_{ij}, in each direction measured in Kbps. The matrix of Figure 9.2 (*b*) has an entry for each source-destination pair. The entry for source *i* to destination *j* shows the route to be used for *i*-*j* traffic, and also the number of packets/sec to be sent from source *i* to destination *j*. For example, 3 packets/sec go from B to D, and they use route BFD to get there. Notice that some routing algorithm has already been applied to derive the routes shown in the matrix.



(a)

		Destination					
		A	B	C	D	E	F
Source	A	■	9	4	1	7	4
	B	9	■	8	3	2	4
	C	4	8	■	3	3	2
	D	1	3	3	■	3	4
	E	7	2	3	3	■	5
	F	4	4	2	4	5	■

(b)

Fig. 9.2 (*a*) A subnet with the capacities shown in Kbps.
(*b*) The traffic in packet/sec. and the routing matrix

Given this information, it is straight-forward to calculate the total in line *i*, λ_i. For example, the B-D traffic contributes 3 packets/sec to the BF line and also 3 packets/sec to the FD line. Similarly, the A-D traffic contributes 1 packets/sec to each of three lines. The total traffic in each eastbound line is shown in the λ_i column of table 9.4. In this example, all the traffic is symmetric, *i.e.*, the XY traffic is identical to the YX traffic for all X and Y. In real networks this condition does not always hold. The figure also shows the mean number of packet/sec on each line, μC_j assuming a mean packet size of $1/\mu = 800$.

The next-to-last column of table 9.4 gives the mean delay for each line derived from the queueing theory formula.

$$T = \frac{1}{\mu C - \lambda}$$

where

$1/\mu$ = is the mean packet size in bits.

C = is the capacity in bps

λ = is the mean flow in packets/sec.

For example:

With a capacity μC = 25 packets/sec and an actual flow λ = 14 packets/sec, the mean delay is gl msec. Note that with λ = 0, the mean delay is still 40 msec, because the capacity is 25 packet/sec. In other words, the "delay" includes with both queueing and service time.

Table 9.4 Analysis of the subnet of Figure 9.2 using a mean packet size of 800-bits. The reverse traffic (BA, CB etc.) is the same as the forward traffic

i	*Line*	λ_i *(pkts/sec)*	C_i *(Kbps)*	μC_i *(pkts/sec)*	T_1 *(msec)*	*Weight*
1	AB	14	20	25	91	0.171
2	BC	12	20	25	77	0.146
3	CD	6	10	12.5	154	0.073
4	AE	11	20	25	71	0.134
5	EF	13	50	62.5	20	0.159
6	FD	8	10	12.5	222	0.098
7	BF	10	20	25	67	0.122
8	EC	8	20	25	59	0.098

To compute the mean delay time for the entire subnet, we take the weighted sum of each of the eight lines, with the weight being the fraction of the total traffic using that line. In this example, the mean turns out to be 86 msec. To evaluate different routing algorithm, we can repeat the entire process, only with different flows to get a new average delay.

9.2.4 Distance Vector Routing

Distance vector routing is a dynamic algorithms. Distance vector routing algorithms operate by having each router maintain a table *i.e.*, vector giving the best known distance to each destination and which line to use to get there. These table are updated by exchanging information with the neighbours.

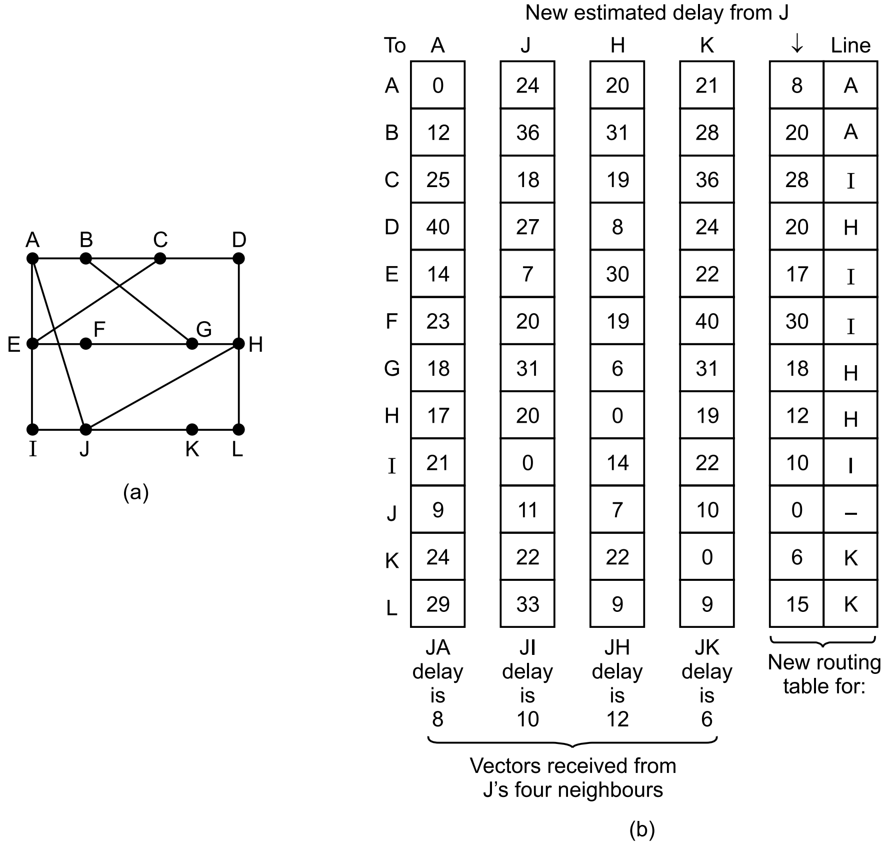
The distance vector routing algorithm is sometimes called by other names, including the distributed Bellman-Ford routing algorithm and the Ford Fulkerson algorithm, after the researchers who developed it. It was the original ARPANET routing algorithm and was also used in the Internet under the name RIP and *i* early versions of DECnet and Novell's IPX. Apple Talk and Cisco routers use improved distance vector protocols.

In distance vector routing, each router maintains a routing table indexed by, and containing one entry for, each router in the subnet. This entry contains two parts: the preferred outgoing line to use for that destination and an estimate of the time of distance to that destination. The metric used might be number of hops, time delay in milliseconds, total number of packets queued along the path, or something similar.

The router is assumed to know the “distance” to each of its neighbours. If the metric is hope the distance is just one hop. If the metric is queue length, the router simply examines each queue. If the metric is delay, the router can measure it directly with special ECHO packets that the receiver just timestamps and sends back as fast as it can.

An example, assume that delay is used as a metric and that the router knows the delay to each of its neighbours. Once every T msec each router sends to each neighbour a list of its estimated delays to each destination.

It also receives a similar list from each neighbour. Imagine that one of these table has just come in from neighbour X, with X_i → m msec via X. By performing this calculation for each neighbour, a router can find out which estimate seems the best and use estimate and corresponding line in this new routing table. Note that the old routing table is not used in the calculation.



(a)

New estimated delay from J

To	A	J	H	K	↓	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K

JA delay is 8 | JI delay is 10 | JH delay is 12 | JK delay is 6

Vectors received from J’s four neighbours

New routing table for:

(b)

Fig. 9.3 (*a*) A subnet, (*b*) Input from A, I, H, K and the new routing table for J.

This updating process is illustrated in Figure 9.3. Part (a) shows a subnet. The first four columns of part (b) shows the delay vectors received from the neighbours of router J. A claims to have a 12 msec delay to B, a 25 msec delay to C, a 40 msec delay to D, etc. Suppose that J has measure or estimated its delay to its neighbours, A, I, H and K as 8, 10, 12 and 6 msec, respectively.

Consider how J computes its new router G. It knows that it can get to A in 8 msec, and A claims to be able to get to G in 18 msec, so J knows it can count on a delay of 26 msec to G if it forwards packets bound from G to A. Similarly, it computes the delay to G via I, H and K as 41 (31 + 10), 18 (6 + 12), and 37 (31 + 6) msec respectively. The best of these value is 18; so it makes an entry in this routing table that the delay to G is 18 msec, and that the route to use is via H. The same calculation is performed for all the other destinations, with the new routing table shown in the last column of the figure.

Distance vector routing works in theory but has a serious drawback in practice: Although it converages to the correct answer, it may do so slowly. In particular, it reacts rapidly to good news, but leisurely to bad news.

9.2.5 Link State Routing

Distance vector routing was used in the ARPANET until 1979, when it was replaced by link state routing. Two primary problems caused its demise.

1. Since the delay metric was queue length, it did not take line bandwidth into account when choosing routes.

   Initially, all the lines were 56 Kbps, so line bandwidth was not an issue, but after some line had been upgraded to 230 Kbps and others to 1.54 Mbps, not taking bandwidth into account was a major problem. Of course, it would have been possible to change the delay metric to factor in line bandwidth.

2. The algorithm often took too long to converge even with tricks like split horizon.

For this reason, it was replaced by an entirely new algorithm now called *link state routing*. Variants of link state routing are now widely used.

The idea behind link state routing is simple and can be stated as five parts. Each router must:

1. Discover its neighbours and learn their network addresses.
2. Measure the delay or cost to each of its neighbour.
3. Construct a packet telling all it has just learned.
4. Send this packet to all other routers.
5. Compute the shortest path to every other router.

In effect the complete topology and all delay are experimentally measured and distributed to every router. Then to find the shortest path to every other router *Dijkstra's algorithm* is used considering five steps.

1. Learning About the Neighbours

When a router is booted, its first task is to learn who its neighbours are. It accomplishes this properly by sending a special HELLO packet on each point-to-point line. The route on the other end is expected to send back a reply telling who it is. These names must be globally unique.

2. Measuring Line Cost

The link state routing algorithm require each router to know, or at least have a reasonable estimate, of the delay to each of its neighbours. The most direct way to determine this delay is to send a special ECHO packet over the line that the other side is required to send back immediate. By measuring the round-trip time and dividing it by two, the sending router can get a reasonable estimate of the delay. For even better results the test can be conducted several times, and the average used.

3. Building Link State Packets

Once the information needed for the exchange has been collected, the next step is for each router to build packet containing all the data. The packet starts with the identity of the sender, followed by a sequence number and age, and a list of neighbours. For each neighbours, the delay to that neighbour is given. An example, subnet is given in Figure 9.4 (a) with delays shown in the lines. The corresponding link state packets for all six routes are shown in Figure 9.4 (b).

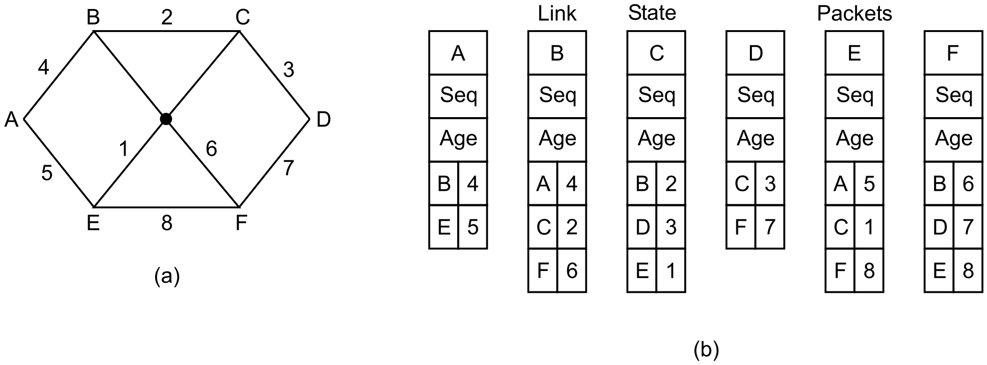


Fig. 9.4 (*a*) A Subnet, (*b*) The link state packets for the subnet

Building the link state packets is easy. The hard part is determining when to build them. Once possibility is to build them periodically, *i.e.*, at regular intervals. Another possibility is when some significant event occurs, such as a line or neighbour going down or coming back up again, or changing its properties appreciably.

4. Distributing the Link State Packets

The trickiest part of the algorithm is distributing the link state packets reliably. As the packets are distributed and installed, the routers getting the first ones will change their routes. Consequently, the different routers may be using different versions of the topology, which can lead to inconsistencies, loops, unreachable machine, and other problems.

5. Computing the New Routes

Once a router has accumulated a full set of link state packets, it can construct the entire subnet graph because every link is represented. Every link is, infact, represented twice, once for each direction. The two values can be averaged or used separately.

Now Dijkstra's algorithm can be run locally to construct the shortest path to all possible destinations. The results of this algorithm can be installed in the routing tables, and normal operation resumed.

9.2.6 Broadcast Routing

For some applications, hosts need to send message to many or all other hosts. *For example,* a service distributing weather reports, stock market updates, or live radio programs might work best by broadcasting to all machines and letting those that are interested read to data "sending a packet to all destinations simultaneously is called broadcasting".

Various methods of broadcast routing have been proposed for doing it.

1. One broadcast method that requires no special features from the subnet is for the source to simply send a distinct packet to each destination.
2. Flooding is another obvious candidate. Although flooding is ill-suited for ordinary point-to-point communication. Here it generates too many packet and consumes too much bandwidth.
3. The third algorithm is multidestination routing. If this method is used, each packet contains either a list of destinations or a bit map indicating the desired destinations. Multidestination routing is like separately addressed packets, except that when several packets must follow the same route, one of them pays full fare and the rest ride free.
4. The fourth broadcasting method, based on the use of a spanning tree. *i.e.*, A spanning tree is connected sub-graph of the network that includes all nodes and has no cycles. It is more communication-efficient than flooding.
5. Our last broadcast algorithm is reverse path forwarding. The advantage of this is that it is both reasonably efficient and easy to implement. It neither require routers to know about spanning tree, nor does it have the overhead of a destination list or bit map in each broadcast packet as does multidestination addressing. Nor does it require any special mechanism to stop the process, as flooding does.

9.3 CONGESTION

When two many packets are present in the subnet, performance degrades. This situation is called **Congestion**. Figure 9.5 depicts the symptom. When the number of packets dumped into the subnet by the hosts is within its carrying capacity. They are all delivered, and the number delivered is proportional to the number sent. However, as traffic increases too far, the routers are no longer able to cope, and they begin losing packets. This tends to make matters worse. At very high traffic, performance collapses completely, and almost no packets are delivered.

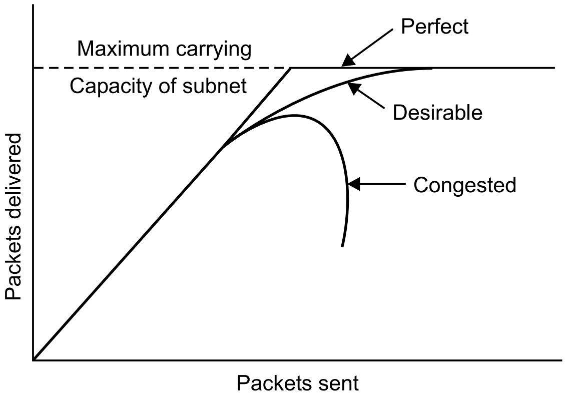


Fig. 9.5 When too much traffic is offered, congestion sets in and performance degrades sharply.

9.3.1 Congestion Control Principles

Many problems in complex systems, such as computer networks, can be viewed from a control theory point of view. This approach leads to dividing all solutions into two groups: open loop and closed loop. Open loop solutions attempt to solve the problem by good design, in essence, to make sure it does not occur in the first place. Once the system is up and running, midcourse corrections are not made.

Tools for doing open loop control include deciding when to accept new traffic, deciding when to discard packets and which ones, and making scheduling decisions at various points in the network. All of these have in common the fact that they make decisions without regard to the current state of the network.

It contrast, closed loop solutions are based on the concept of a feedback loop. This approach has *three parts* when applied to congestion control:

1. Monitor the system to detect when and where congestion occurs.
2. Pass this information to places where action can be taken.
3. Adjust system operation to correct the problem.

Various metrics can be used to monitor the subnet for congestion. Chief among these are the percentage of all packets discarded for lack of buffer space, the average queue length, the number of packets that time out and are retransmitted, the average packet delay, and the standard deviation of packet delay. In all cases, rising numbers indicate growing congestion.

The second step in the feedback loop is to transfer the information about the congestion from the point where it is detected to the point where something can be done about it. The obvious way is for the router detecting the congestion to send a packet to the traffic sources, announcing the problem. Of course, these extra packets increase the load at precisely the moment that more load is not needed, namely, when the subnet is congested.

However, other possibilities also exist. For example, a bit or field can be reserved in every packet for routers to fill in whenever congestion gets above some threshold level.

When a router detects this congested state, it fills in the field in all outgoing packets, to warn the neighbours.

Still another approach is to have hosts or routers send probe packets out periodically to explicitly ask about congestion. This information can then be used to route traffic around problem areas. Some radio stations have helicopters flying around their cities to report on road congestion in the hope that their listeners will route their packets (cars) around hot spots.

In all feedback schemes, the hope is that knowledge of congestion will cause the hosts to take appropriate action to reduce the congestion. To work correctly, the time scale must be adjusted carefully. If every time two packets arrive in a row, a router yells **STOP** and every time a router is idle for 20 µsec it yells **GO**, the system will oscillate wildly and never converge. On the other hand, if it waits 30 minutes to make sure before saying anything, the congestion control mechanism will react too sluggishly to be of any real use. To work well, some kind of averaging is needed, but getting the time constant right is a non-trivial matter.

Many congestion control algorithms are known. To provide a way to organize them in a sensible way, **Yang and Reddy** (1995) have developed a taxonomy for congestion control algorithms. They begin by dividing all algorithms into open loop or closed loop, as described above. They further divide the open loop algorithms into ones that act at the source versus ones that act at the destination. The **closed loop algorithms** are also divided into two subcategories.

- **Explicit feedback** versus **implicit feedback**.
- In **explicit** feedback algorithms, packets are sent back from the point of congestion to warn the source.
- In **implicit** algorithms, the source deduces the existence of congestion by making local observations, such as the time needed for acknowledgements to come back.

The presence of congestion means that the load is (temporarily) greater than the resources can handle. Two solutions come to mind : increase the resources or decrease the load. For example, the subnet may start using dialup telephone lines to temporarily increase the bandwidth between certain points.

However, sometimes it is not possible to increase the capacity, or it has already been increased to the limit. The only way then to beat back the congestion is to decrease the load.

Some of these methods, which we will study shortly, can best be applied to virtual circuits. For subnets that use virtual circuits internally, these methods can be used at the network layer. For datagram subnets, they can nevertheless sometimes be used on transport layer connections.

9.3.2 Congestion Prevention Policies

Given table 9.5 shows different data link layer, network, and transport policies that can affect congestion.

Table 9.5 Policies that affect congestion

Layer	*Policies*
Transport	• Retransmission policy • Out-or-order caching policy • Acknowledgement policy • Flow control policy • Timeout determination
Network	• Virtual circuits versus datagram inside the subnet • Packet queueing and service policy • Packet discard policy • Routing algorithm • Packet lifetime management
Data link	• Retransmission policy • Out-or-order caching policy • Acknowledgement policy • Flow control policy

Let us start at the data link layer and work our way upward. The **retransmission policy** deals with how fast a sender times out and what it transmits upon timeout. A jumpy sender that times out quickly and retransmits all outstanding packets using *go* back *n* will put a heavier load on the system than a leisurely sender that uses selective repeat. Closely related to this is cashing policy. If receivers routinely discard their destination, thus inducing retransmissions.

In the transport layer, the same issues occur as in the data link layer, but in addition determining the timeout interval is harder because the transit time across the network is less predictable than the transit time over a wire between two routers. If it is too short, extra packets will be sent unnecessarily. If it is too long, congestion will be reduced, but the response time will suffer whenever a packet is lost.

9.3.3 Leaky Bucket Algorithms

Imagine a bucket with a small hole in the bottom, as shown in Figure 9.6 (a). No matter at what rate water in the bucket will overflow and the outflow is at a constant rate, P, when there is any water in the bucket, and zero when the bucket is empty. Also, once the bucket is full, any additional water entering it spills over the sides and is lost.

The same idea can be applied to packets, as shown in Figure 9.6 (*b*). Conceptually, each host is connected to the network by an interface containing a leaky bucket, that is a finite internal queue. If a packet arrives at the queue when it is full, the packet is discarded. In other words, if one or more processes within the host try to send a packet is unceremoniously discarded. This arrangement can be built into the hardware interface or simulated by the host operating system. It was first proposed by Turner (1986) and is called the **leaky bucket algorithm.** In fact, it is nothing other than a single-server queuing system with constant service time.

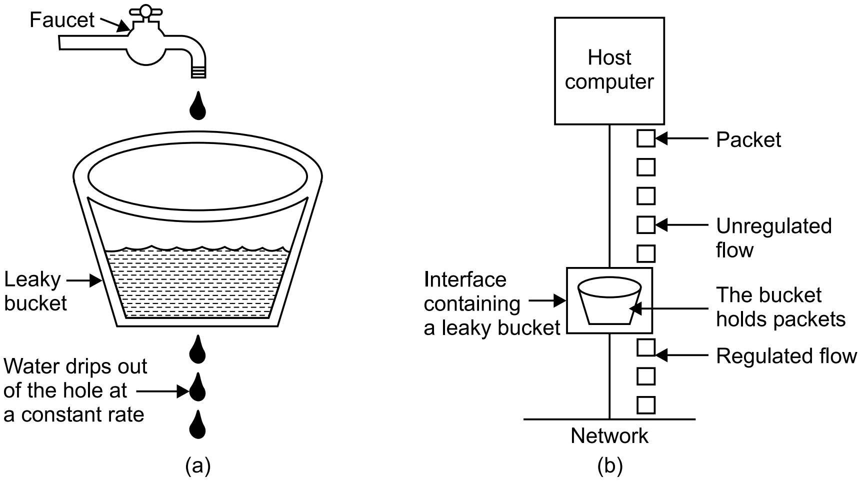


Fig. 9.6 (*a*) A leaky bucket with water (*b*) A leaky bucket with packets

The host is allowed to put one packet per clock tick onto the network. Again, this can be enforced by the interface card or by the operating system. This mechanism turns an uneven flow of packet from the user processes inside the host into an even flow of packets onto the network, smoothing out bursts and greatly reducing the chances of congestion.

When the packets are all the same size, this algorithm can be used as described. However, when variable-sized packets are being used, it is often better to allow a fixed number of bytes per tick, rather than just one packet. Thus if the rule is 1024-bytes per tick, a single 1024-bytes packet can be admitted on a tick, two 512-byte packets, four 256-byte packets, and so on. If the residual byte count is too low, the next packet must wait until the next tick.

Implementing the original leaky bucket algorithm is easy, the leaky bucket consists of a finite queue. When a packet arrives, if there is room on the queue it is appended to the queue, otherwise, it is discarded. At every clock tick, one packet is transmitted.

The byte-counting leaky bucket is implemented almost the same way. At each tick, a counter is initialized to *n*. If the first packet on the queue has fewer bytes than the current value of the counter, it is transmitted, and the counter is decremented by that number of bytes. Additional packets may also be sent, as long as the counter is high enough. When the counter drops below the length of the next packet on the queue, transmission stops until the next tick, at which time the residual byte count is overwritten and lost.

9.3.4 Token Bucket Algorithm

The **leaky bucket algorithm** enforces a rigid output pattern at the average rate, no matter how bursty the traffic is. For many applications, it is better to allow the output to speed up somewhat when large bursts loses data. One such algorithm is the token bucket algorithm, the leaky bucket holds tokens, generated by a clock at the rate of one token every ΔT sec. In Figure 9.7 (*a*) we see a bucket holding three tokens, with five packets waiting to be transmitted. For a packet to be transmitted, it must capture and destroy one token. In Figure 9.7 (*b*) we see that three of the five packets have got through, but the other two are stuck waiting for two more tokens to be generated.

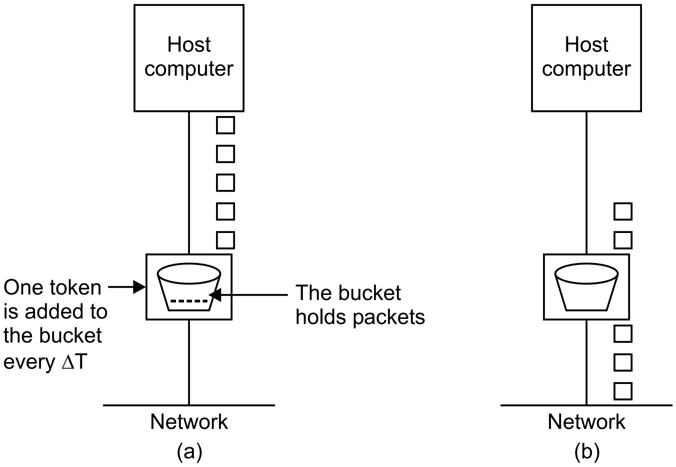


Fig. 9.7 The token bucket algorithm (*a*) Before (*b*) After

The **token bucket algorithm** provides a different king of traffic shaping than the leaky bucket algorithm does not allow idle hosts to save up permission to send large bursts later. The token bucket algorithm does allow saving, up to the maximum size of the bucket, *n*. This property means that bursts of up to *n* packets can be sent at one, allowing some burstiness in the output stream and giving faster response to sudden bursts of input.

Another difference between the two algorithms is that the token bucket algorithm throws away tokens when the bucket fills up but never discards packets. In constant, the leaky bucket algorithm discards packets when the bucket fills up.

Here too, a minor variant is possible, in which each token represents the right to send not one packet, but K bytes. A packet can only be transmitted if enough tokens are available to cover its length in bytes. Fractional token are kept for future use.

The leaky bucket and token bucket algorithms can also be used to smooth traffic between routers, as well as being used to regulate host output as in our example. However, one clear difference is that a token bucket regulating a host can make the host stop sending when the rules say it must. Telling a router to stop sending while its input keeps pouring in may result in lost data.

The implementation of the basic token bucket algorithm is just a variable that counts tokens. The counter is incremented by one very ΔT and decremented one whenever a packet is sent. When the counter hits zero, no packets may be sent. In the byte-count variant, the counter is increment by K bytes every ΔT and decremented by the length of each packet sent.

Essentially what the token bucket does is allow bursts, but up to a regulated maximum length.

9.3.5 Choke Packets

Let us now turn to an approach that can be used in both virtual circuit and datagram subnets. Each router can easily monitor the utilization of its output lines and other resources. For example, its can associate with each line a real variable, μ whose value, between 0.0 and 1.0, reflects the recent utilization of that time. To maintain a good estimate of μ, a sample of the instantaneous line utilization, *f* (either 0 or 1), can be made periodically and μ updated according to

$$\boxed{\mu_{new} = a\mu_{old} + (1 - a)\, f}$$

where, a determines how fast the router forest recent history.

Whenever μ moves above the threshold, the output line enters a “warning” state. Each newly arriving packet is checked to see if its output line is in warning state. If so, the router sends a **choke packet** back to the source host, giving it the destination found in the packet. The original packet is tagged (a header bit is turned on) so that it will not generate any more choke packets further along the path and is then forwarded in the usual way.

When the source host gets the choke packet, it is required to reduce the traffic sent to the specified destination by X per cent. Since other packets aimed at the same destination are probably already under way and will generate yet more choke packets, the host should ignore choke packets referring to that destination for a fixed time interval. After that period has expired, the host lists for more choke packets for another interval. If one arrives, the line is still congested, so the host reduces the flow still more and begins ignoring choke packets again. If no choke packets arrive during the listening period, the host may increase the flow again the feedback implicit in this protocol can help prevent congestion yet not throttle any flow unless trouble occurs.

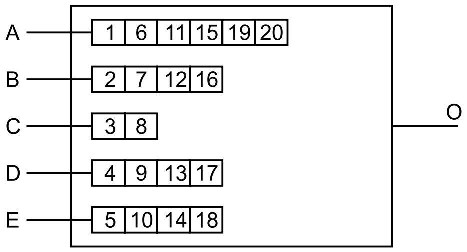
Hosts can reduce traffic by adjusting their policy parameters, *for example*, window size or leaky bucket output rate. Typically, the first choke packet causes the data rate to be reduced to 0.50 of its previous rate, the next one causes a reduction to 0.25, and so on. Increases are down in smaller increments to prevent congestion from reoccurring quickly.

Weighted Fair Queuing

A problem with using packets is that the action to be taken by the source hosts is voluntary. Suppose that a router is being swamped by packets from four sources, and it sends choke packets to all of them. One of them cuts back, as it is supposed to, but the other three just keep blasting away. The result is that the honest host gets an even smaller share of the bandwidth than it had before.

To get ground this problem, and thus make compliance more attractive. **Nagle** proposed a **fair queuing algorithm**. The essence of the algorithm is that routers have multiple queues for each output line, one for each sources. When a line becomes idle, the router scans the queues round robin, taking the first packet on the next queue. In this easy, with hosts competing for a giver output line, each hosts goes to send one out of every n packets. Sending more packets will not improve this fraction. Some **ATM** switches use this algorithm.

Although a start, the algorithm has a problem : it gives more bandwidth to hosts that use large packets than to hosts that use small packets. Demers suggested an improvement in which the round robin is done in such a way as to simulate a byte-by-byte round robin, instead of a packet-by-packet round robin. In effect, it scans the queues repeatedly, byte-for-byte, until it finds the tick on which each packet will be finished. The packets are then sorted in order of their finishing and sent in that order. The algorithm is shown in Figure 9.8.



(a)

Packet	Finishing time
C	8
B	16
D	17
E	18
A	20

(b)

Fig. 9.8 (*a*) A router with five packets queued for line 0, (*b*) Finishing times for the five packets

In Figure 9.8 (*a*) we see packets of length 2 to 6 bytes. At (virtual) clock tick 1, the first byte of the packet on line A is sent. Then goes the first byte of the packet on line B, and so on. The first packed to finish is C, after eight ticks. The sorted order is given in Figure 9.8 (*b*). In the absence of new arrivals, the packets will be sent in the order listed, from C to A.

One problem with this algorithm is that it gives all hosts the same priority. In many situations, it is desirable to given the file and other servers more bandwidth than clients, so they can be given two or more bytes per tick. This modified algorithm is called **weighted fair queuing** and is widely used. Sometimes the weight is equal to the number of virtual circuits or flows coming out of a machine, so each process gets equal bandwidth.

Hop-by-Hop Choke Packets

At high speeds and over long distances, sending a choke packet to the source hosts does not work well because the reaction is so slow. The effect of this hop-by-hop scheme is to provide quick relief at the point of congestion at the price of using up more buffers upstream. This is illustrated as shown in Figure 9.9 that congestion can be nipped in the bud without losing any packets. For example, a host in San Francisco (router A in Figure 9.9 (*a*) that is sending traffic to a host in New York (router D in Figure 9.9 (*a*) at 155 Mbps.) If the New York host begins to run out of buffers it will take about 30 msec for a choke packet to get back to San Francisco to tell it to slow down. The choke packet propagation as the second, third, and fourth steps is given in Figure 9.9 (*a*). In those 30 msec, another 4.6 megabits will have been sent. Even if the host in San Francisco completely shuts down immediately, the 4.6 megabits in the pipe will continue to pur in and have to be dealt with. Only in the seventh diagram in Figure 9.9 (*a*) will the New York router notice a slower flow.

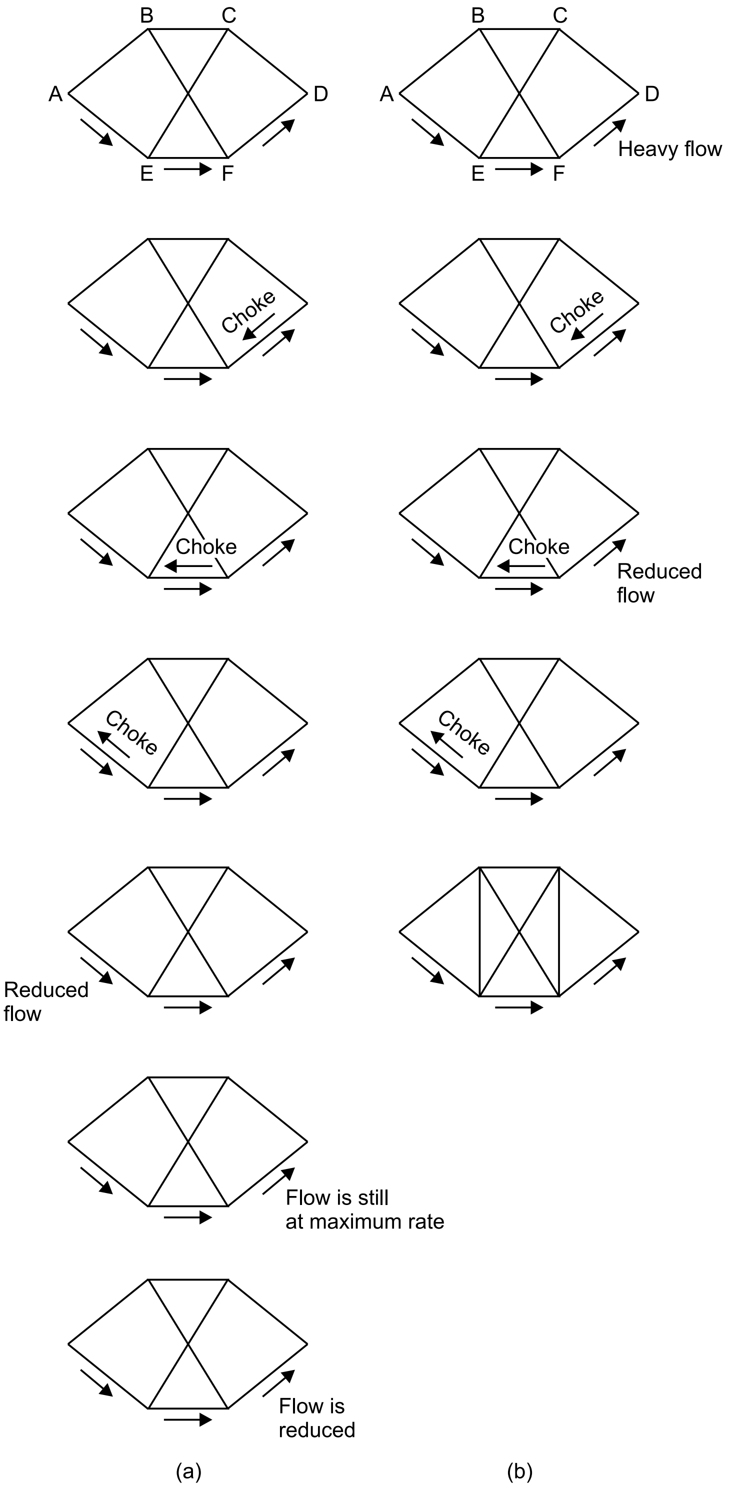


Fig. 9.9 (*a*) A choke packet that affects only the source, (*b*) A choke packet that affects each hop it passes through

An alternative approach is to have the choke packet take effect at every hop it passes through, as shown in the sequence of Figure 9.9 (b). Here, as soon as the choke packet reaches F, F is required to reduce the flow to D. Doing so will require F to devote more buffers to the flow, since the source is still sending away at full blast, but it gives D immediate relief, choke packet reaches E, which tells E to reduce the flow to F. This action puts a greater demand on E's buffers but gives F immediate relief. Finally, the choke packet reaches A and the flow genuinely shows down.

9.3.6 Load Shedding

When none of the above methods make the congestion disappear, routers can bring out the heavy artillery: load shedding. **Load shedding** is a fancy way of saying that when routers are being inundated by packets that they cannot handle, they just throw them away. The term comes from the world of electrical power generation where it refers to the practice of utilities intentionally blacking out certain area to save the entire grid from collapsing on hot summer days when the demand for electricity greatly exceeds the supply.

A router drowning in packets can just pick packets at random to drop, but usually it can do better than that. Which packet to discard may depend on the applications running. For file transfer, an old packet is worth more than a new one because dropping packet 6 and keeping packets 7 through 10 will cause a gap at the receiver that may force packets 6 through 10 to be retransmitted. In a 12 packet file, dropping 6 may require 7 through 12 to be retransmitted, whereas dropping 10 may require only 10 through 12 to be retransmitted. In contrast, for multimedia, a new packet is more important than an old one. The former policy is often called **win** and the latter is often called **milk**.

A step above this in intelligence requires co-operation from the senders. For many applications, some packets are more important than others. For example, certain algorithms for compressing video periodically transmit and entire frame and then send subsequent frames as differences from the last full frame. In this case, dropping a packet that is part of a difference is preferable to dropping one that is part of a full frame.

To implement an intelligent discard policy, applications must mark their packets in priority classes to indicate how important they are. If they do this, when packets have to be discarded, routers can first drop packets from the lowest class, then the next lowest class, and so on. Of course, unless there is some significant incentive to mark packets as anything other than **VERY IMPORTANT-NEVER**. **EVER DISCARD**, nobody will do it.

The incentive might be in the form of money, this the low-priority packets being cheaper to send than the high-priority ones. Alternatively, priority classes could be coupled with traffic shaping.

Another option is to allow hosts to exceed the limits specified in the agreement negotiated when the virtual circuit was set up, but subject to the condition that all excess traffic be marked as low-priority. Such a strategy is actually not a bad idea, because it makes more efficient use of idle resources, allowing hosts to use them as long as nobody else is interested, but without establishing a right to them when times get tough.

Making packets by class requires one or more header bits in which to put the priority. ATM cells have 1-bit reserved in the header for this purpose, so every **ATM** cell is labeled either as low-priority or high-priority. **ATM** switches indeed use this bit when making discard decisions.

9.3.7 Jitter Control

For applications such as audio and video transmission, it does not matter much if the packets take 20 msec or 30 msec to be delivered, as long as the transit time is constant. Having some packets taking 20 msec and others taking 30 msec will give an uneven quality to the sound or image. Thus the agreement might be that 99 per cent of the packets be delivered with a delay in the range of 24.5 msec to 25.5 msec. The mean value chosen must be feasible, of course. In other words, an average amount of congestion must be calculated in.

The jitter can be bounded by computing the expected transit time for each hop along the path. When a packet arrives at a router, the router checks to see how much the packet is behind or ahead of its schedule. This information is stored in the packet and updated at each hop. If the packet is ahead of schedule, it is held just long enough to get it back on schedule. If it is behind schedule, the router tries to get it out the door quickly. In fact, the algorithm for determining which of several packets competing for an output line should go next can always choose the packet furthers behind in its schedule. In this way, packets that are ahead of schedule get slowed down and packets that are behind schedule get speeded up, in both cases reducing the amount of jitter.

QUESTIONNARIES

1. Explain services provided by network layer to transport layer.
2. Compair the virtual circuit and datagram subnet.
3. What are adaptive and non-adaptive algorithm.
4. State and explain optimal Routing.
5. Explain shortest path algorithm.
6. Explain distance vector routing. What is count-to-infinity problem? How it is reduced?
7. Explain Belman-Ford routing algorithm.
8. Explain split horizon back.
9. Discuss behaviour of the subnets under different traffic loads.
10. What is the difference between congestion control and flow control?
11. Explain two approaches for congestion control.
12. State and explain congestion preventation policies implemented at different layers.
13. What is traffic shaping.
14. Discuss causes of congestion and methods to control congestion. Explain token bucket algorithm.
15. What is congestion? Explain leaky bucket algorithm.
16. Explain load shedding and jitter control.
17. Write short note on:
    - Flooding
    - Link state routing
    - Choke packets
    - Flow based routing
    - Hierarchical routing

Chapter 10

INTERNETWORKING

INTRODUCTION

Interconnection of individual network provides users with an increased level of connectivity, resource sharing and application-to-application communication potential. This typically requires the installation of additional hardware and software. This chapter presents a discussion of the principles underlying network interconnection technology.

In general, interconnecting networks requires the use of relays. Data origination in one network and destined for a receiver on another network must traverse one or more relays. Because interconnected networks may be dissimilar, the relay typically performs any translation functions required to make the data originating in one network compatible with the other network.

10.1 INTERNETWORKING DEVICES

There are various **internetwork devices** which are given here:

1. Gateway

It is a device that connects two or more dissimilar networks. Transport gateways make a connection between two networks at the transport layer. Application gateways connect two parts of an application in the application layer. Gateways may be used either in full gateway form or two half-gateways form.

2. Repeater

Repeaters are low-level devices that just amplify or regenerate weak signals.

3. Bridge

Bridges are store and forward devices, there are various types of bridges such as transparent bridges, remote bridges, spanning tree and source routing bridges.

4. Multiprotocol router

They are conceptually similar to bridges, except that they are found in the network layer. They just take incoming packets from one line and forward them on another, just as all routers do, but the lines may belongs to different networks and use different protocols.

10.2 CONCATENATED VIRTUAL CIRCUITS

Two styles of internetworking are common: a connection-oriented concatenation of virtual circuit subnets, and a datagram internet style. We will now examine these in turn. In the concatenated virtual circuit model, shown in Figure 10.1, a connection to a host in a distant network is set up in a way similar to the way connections are normally established. The subnet sees that the destination is remote and builds a virtual circuit to the router nearest the destination network. Then it constructs a virtual circuit from that router to an external "gateway". This gateway records the existence of the virtual circuit in its tables and proceeds to build another virtual circuit to a router in the next subnet. This process continues until the destination host has been reached.

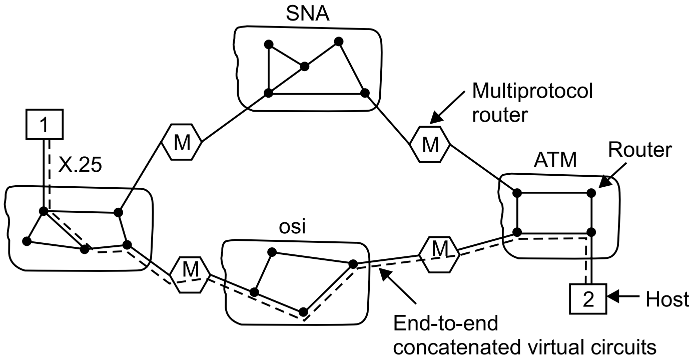


Fig.10.1 Internetworking using concatenated virtual circuits

Once data packets begin flowing along the path, each gateway relays incoming packets. Converting between packet formats and virtual circuit numbers as needed. Clearly, all data packets must traverse the same sequence of gateways, and thus arrive in order.

The essential feature of this approach is that a sequence of virtual circuits is setup from the source through one or more gateways to the destination. Each gateway maintains tables telling which virtual circuits pass through it, where they are to be routed, and what the new virtual circuit number is.

Although Figure 10.1 shows the connection made with a full gateway, it could equally well be done with half gateways.

This scheme works best when all the networks have roughly the same properties for example, if all of them guarantee reliable delivery of network layer packets, then barring a crash somewhere along the route, the flow from source to destination will also be reliable. Similarly, if none of them guarantee reliable delivery, then the concatenation of the virtual circuits is not reliable either. On the other hand, if the source machine is on a network that does guarantee reliable delivery, but one of the intermediate networks can lose packets, the concatenation has fundamentally changed the nature of the service.

Concatenated virtual circuits are also common in the transport layer. In particular, it is possible to build a bit pipe using, say, OSI, which terminates in a gateway, and have a TCP connection go from the gateway to the next gateway. In this manner, an end-to-end virtual circuit can be built spanning different networks and protocols.

10.3 CONNECTIONLESS INTERNETWORKING

The alternative internet work model is the datagram model, shown in Figure 10.2. In this model, the only service the network layer offers to the transport layer is the ability to inject datagrams into the subnet and hope for the best. There is no notion of a virtual circuit at all in the network layer. Let along a concatenation of them. This model does not require all packets belonging to one connection to transverse the same sequence of gateways. In Figure 10.2 datagrams from host 1 to host 2 are shown taking different routes through the internetwork. A routing decision is made separately for each packet is sent. This strategy can use multiple routes and thus achieve a higher bandwidth than the concatenated virtual circuit model. On the other hand, there is no guarantee that the packets arrive at the destinations in order, assuming that they arrive at all.

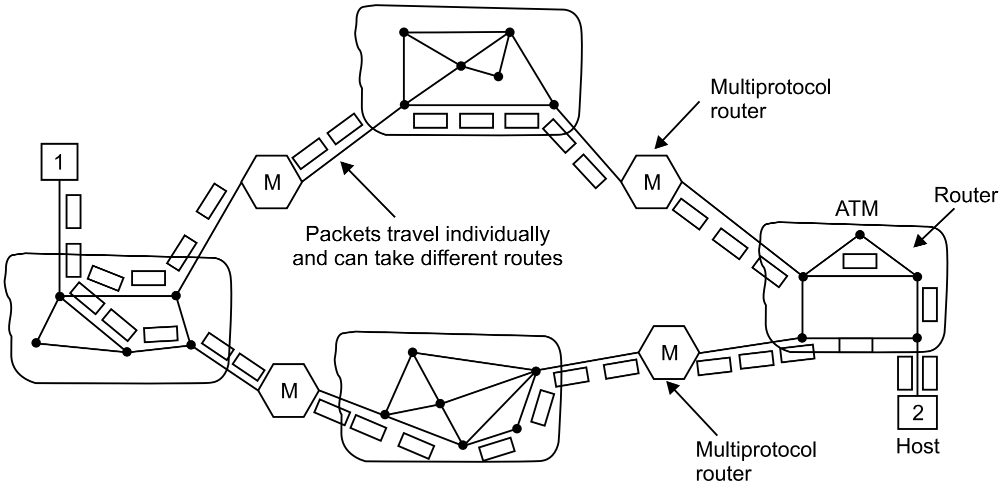


Fig. 10.2 A Connectionless internet

The mode of Figure 10.2 is not quite as simple as it looks. For one thing, if each network has its own network layer protocol, it is not possible for a packet from one network to transit another one. One could imagine the multiprotocol routers actually trying to translate from one format to another, but unless the two formats are close relatives with the same information fields, such conversions will always be incomplete and aften demand to failure. For this reason, conversion is rarely attempted.

A second and more serious problem, is addressing. Imagine a simple case: a host on the Internet is trying to send an IP packet to a host on an adjoining OSI host. The OSI datagram protocol. CLNP, was base on IP and is close enough to it that a conversion might work well.

The trouble is that IP packets carry the 32-bit Internet address of the destination host in a header field. OSI hosts do not have 32-bit Internet addresses. They use decimal addresses similar to telephone numbers.

To make it possible for the multiprotocol router to convert between formats, someone would have to assign a 32-bit Internet address to each OSI host. Taken to the limit, this approach would mean assigning an Internet address to every machine in the world that an Internet host might want to talk to. It would also mean assigning and OSI address to every machine in the world that an OSI host might want to talk to. The same problem occurs with every other address space (SNA, Apple Tak, etc.). The problems here are insurmountable. In addition, someone would have to maintain a database mapping everything to everything.

Another idea is to design a universal "internet" packet and have all routers recognize it. This approach is, in fact, what **IP** is ... a packet designed to be carried through many networks. The only problem is that IPX, CLNP and other "universal" packets exist too, making all of them less than universal. Getting everybody to agree to a single format is just not possible.

Let us now briefly recap the two ways internetworking can be attacked. The concatenated virtual circuit model has essentially the same advantages as wing virtual circuits within a single subnet: buffers can be reserved in advance, sequencing can be guaranteed, short headers can be used, and the troubles caused by delayed duplicate packets can be avoided.

It also has the same **disadvantages** : table space required in the routers for each open connection, no alternate routing to avoid congested areas, and vulnerability to router figures along the path. It also has the disadvantage to being difficult, if not impossible, to implement if one of the networks involved an unreliable datagram network.

10.4 TUNNELING

Handling the general case or making two different networks interwork is exceedingly difficult. However, there is a common special case that is manageable. This case is where the source and destination hosts are on the same type of network, but there is a different network in between. As an example, think of an international bank with a TCP/IP based Ethernet in Paris, a TCP/IP based Ethernet in London, and a **PTTWAN** in between, as shown in Figure 10.3.

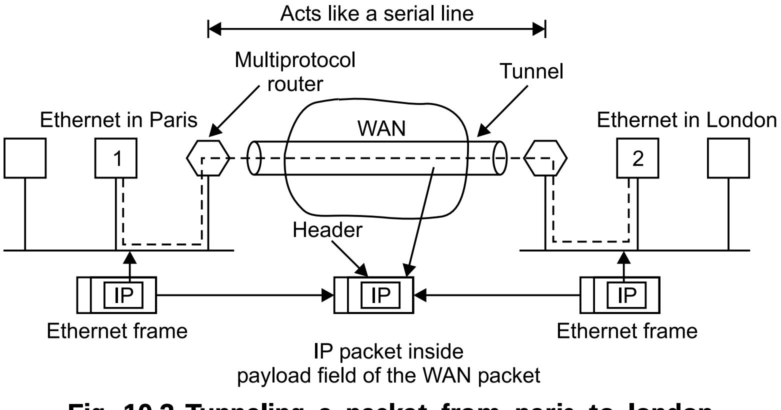


Fig. 10.3 Tunneling a packet from paris to london

The solution to this problem is a technique called **tunneling.** To send an IP packet to host 2, host 1 constructs the packet containing the IP address of host 2, inserts it into an Ethernet frame addressed to the Paris multiprotocol router and the IP packet, inserts it in the payload field of the WAN network layer packet, and addresses the latter to the WAN address of the London multiprotocol router. When it gets there, the London router removes the IP packet and sends it to host 2 inside an Ethernet frame.

The WAN can be seen as a big tunnel extending from one multiprotocol router to the other. The IP packet just travels from one end of the tunnel to the other, snug in its nice box. It does not have to worry about dealing with the WAN at all neither do the host on either Ethernet. Only the multiprotocol route has to understand IP and WAN packets. In effect the entire distance from the middle of one multiprotocol router to the middle of the other acts like a serial line.

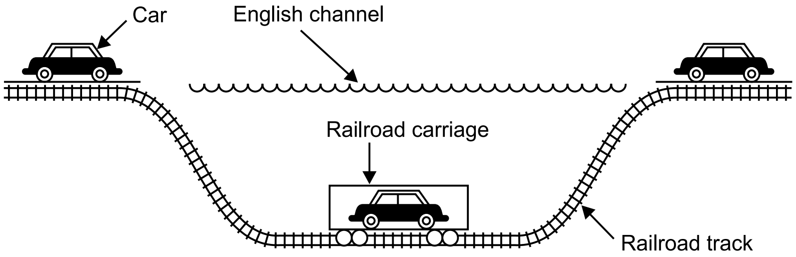


Fig. 10.4 Tunneling a car from France to England

Although an analogy may make tunneling clearer. Consider a person driving her Car from Paris to London. Within France, the car moves under its own power, but when it hits the English channel, it is loaded into a high-speed train and transported to England through the channel. Effectively, the car is being carried as freight, as depicted in Figure 10.4. At the far end, the car is let loose on the English roads and once again continues to move under its own power. Tunneling of packets through a foreign network works the same way.

10.5 INTERNETWORK ROUTING

Routing through an internetwork is similar to routing within a single subnet, but with some added complications. Consider, for example, the internetwork of Figure 10.5 (a) in which five networks are connected by six multiprotocol routers. Making a graph model of this situation is complicated by the fact that every multiprotocol router can directly access (*i.e.*, send packets to) every other route connected to any network to which it is connected. For example, B in Figure 10.5 (a) can directly access A and C via network 2 and also D via network 3. This leads to the graph of Figure 10.5 (b).

Once the graph has been constructed, know routing algorithms, such as the distance vector and link state algorithms, can be applied to the set of multiprotocol routers. This gives a two-level routing algorithm: within each network an **interior gateway protocol** is used, but between the networks, and **exterior gateway protocol** is used (“gateway” is an older term for “router”). In fact, since each network is independent, they may all use different algorithms. Because each network in an internetwork is independent of all the others, it is often referred to as an Autonomous System (AS).

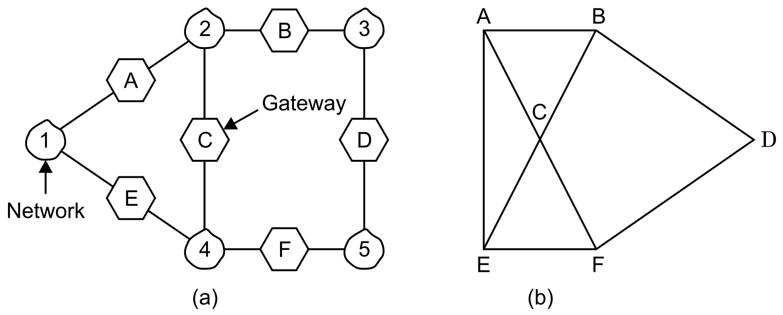


Fig. 10.5 (a) An internetwork (b) A graph of the internetwork

A typical internet packet starts out on its LAN addressed to the local multiprotocol router. After it gets there, the network layer code decides which multiprotocol router to forward the packet to, using its own routing tables. If that router can be reached using the packet's native network protocol, it is forwarded there directly. Otherwise it is tunneled there, encapsulated in the protocol required by the intervening network. This process is repeated until the packet reaches the destination network.

One of the differences between internet work routing and internetwork routing is that internetwork routing often requires crossing international boundaries. Various laws suddenly come into play, such as Sweden's strict privacy laws about exporting personal data about Swedish citizens from Sweden. Another example is the Canadian law saying that data traffic originating in Canada and ending in Canada may not leave the country. This law means that traffic from Windows, Ontario to Vancouver may not be routed via nearby Detroit.

Another difference between interior routing is the cost. Within a single network, a single charging algorithm normally applies. However, different networks may be used different managements, and one route may be less expensive than another. Similarly, the quality of service offered by different networks may be different, and this may be a reason to choose one route over another.

10.6 FRAGMENTATION

Each network imposes some maximum size on its packets. These limits have various causes, among them:

1. **Hardware** (*e.g.*, the width of a TDM transmission slot)
2. **Operating System** (*e.g.*, all buffers are 512-bytes.)
3. **Protocols** (*e.g.*, the number of bits in the packet length field)
4. **Compliance** with some (inter) national standard.
5. Desire to **reduce error** induced retransmission to some level.
6. Desire to **prevent one packet** from occupying the channel too long.

The result of all these factors is that the network designers are not free to choose any maximum packet size they wish. Maximum payloads range from 48-bytes (**ATM Cells**) to 65,515 bytes (IP Packets), although the payload size in higher layers is often larger.

An obvious problem appears when a large packet wants to travel through a network. Whose maximum packet size is too small. One solution is to make sure the problem does not occur in the first place. What happens if the original source packet is too large to be handled by the destination network? The routing algorithm can hardly bypass the destination.

Basically, the only solution to the problem is to allow gateways to break packets up into **fragments**, sending each fragment as a separate internet packet. However, as every parent of a small child knows, converting a large object into small fragments is considerably easier than the reverse process. Packet-switching networks, too, have trouble putting the fragment back together again.

In the ATM world, fragmentation is called **segmentation**, the concept is the same, but some of the details are different. Fragmentations are of following types:

Transparent Fragmentation

The first strategy is to make fragmentation caused by a "small packet" network transparent to any subsequent networks through which the packet m1ust pass on its way to the ultimate destination. It is simple but has some problems. For one thing, the exit gateway must know when it has received all the pieces, so that either a count field or an "end of packet" bit must be included in each packet. For another thing, all packets must exit via the same gateway. By not allowing some fragments to follow one router to the ultimate destination, and other fragments a disjoint route, some performance may be lost. A last problem is the overhead required to repeatedly reassemble and then refragment a large packet passing through a series of small packet network.

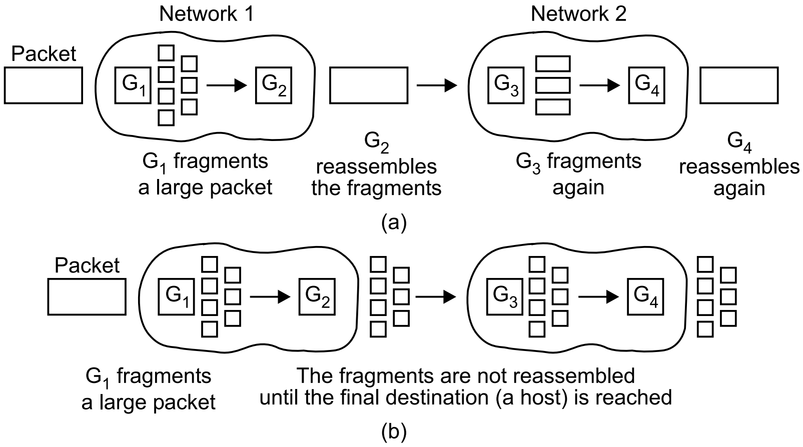

Network 1
Network 2
Packet
G_1
G_2
G_3
G_4
G_1 fragments a large packet
G_2 reassembles the fragments
G_3 fragments again
G_4 reassembles again
(a)
Packet
G_1
G_2
G_3
G_4
G_1 fragments a large packet
The fragments are not reassembled until the final destination (a host) is reached
(b)

Fig. 10.6 (*a*) Transparent fragmentation (*b*) Non-transparent fragmentation

Non-transparent Fragmentation

The strategy is to refrain from recombining fragments at any intermediate gateways. Once a packet has been fragmented, each fragment is treated as though it were an original packet. All fragments are passed through the exist gateway, as shown in Figure 10.6 (b).

Non-transparent fragmentation also has some problems. For example, it requires every host to be able to do reassembly. Yet another problem is that when a large packet is fragmented the total overhead increases, because each fragment must have a header. In this method the overhead remains for the rest of the journey. An **advantage** of this method however, is that multiple exist gateways can now be used and higher performance can be achieved. Of course, if the concatenated virtual circuit model is being used, this advantage is of no use.

10.7 FIRE WALLS

The ability to connect any computer, anywhere, to any other computer. Anywhere, is a mixed blessing for individuals at home, wandering around the Internet is lots of fun. For corporate security managers, it is a nightmare. Most companies have large amounts of confidential information on-line trade secrets, product development plans, marketing strategies, financial analysis, etc. Disclosure of this information to a competitor could have consequences.

In addition to the danger of information leaking out, there is also a danger of information leaking in. In particular, viruses, worms and other digital pests can breach security, destroy valuable data, and waste large amounts of administrators time trying to clean up the mess they leave, often they are imported by careless employees who want to play some new game.

Consequently, mechanisms are needed to keep "good" bits in and "bad" bits out. One method is to use encryption. This approach protects data in transit between secure sites.

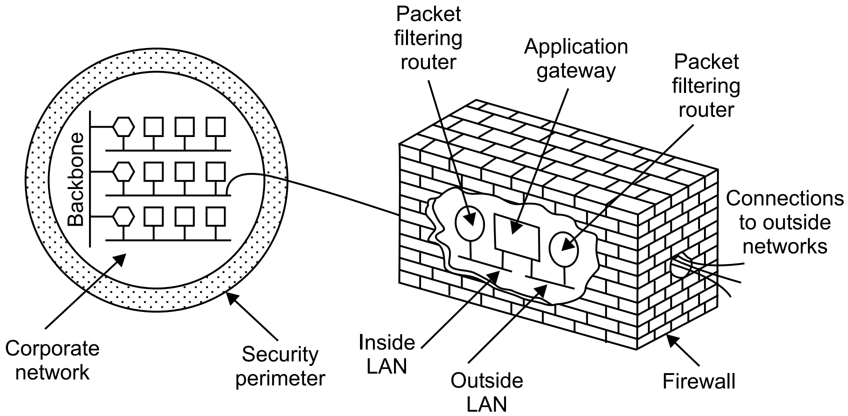


Fig. 10.7 A firewall consisting of two packet filters and an application gateway

Firewalls are just a modern adaption of that old medical security standby: digging a deep moat around your castle. This design forced everyone entering or leaving the castle to pass over a single drawbridge, where they could be inspected by the I/O police. With network the same trick is possible : a company can have many LANs connected in arbitrary ways, but all traffic to or from the company is forced through an electronic drawbridge (firewall), as shown in Figure 10.7.

The firewall in this configuration has two components: two routers that do packet filtering and an application gateway. Simpler configuration also exist, but the advantage of this design is that every packet must transit two filters and an application gateway to go in or out. No other route exists. Readers who think that one security checkpoint is enough clearly have not made an international flight on a scheduled airline recently.

Each **packet filter** is a standard router equipped with some extra functionality. The extra functionality allow every incoming or outgoing packet to be inspected packets meeting some criterion are forwarded normally. Those that fail the test are dropped.

In Figure 10.7, most likely the packet filter on the inside LAN checks outgoing packets and the one on the outside LAN checks incoming packets. Packets crossing the first hurdle go to the application gateway for further examination. The point of putting the two packet filters on different LANs is to ensure that no packet gets in or out without having to pass through the application gateway: there is no path around it.

Packet filters are typically driven by tables configured by the system administrator. These tables list sources and destinations that are acceptable, sources and destinations that are blocked, and default rules about what to do with packets coming from or going to other machines.

In the common case of a UNIX setting, a source or destination consists of an IP address and a port. Ports indicate which service is desired. For example, port 23 is for Telnet, port 79 is for Finger, and port 119 is for USENET news. A company could block incoming packets for all IP addresses combined with one of these ports. In this way, no one outside the company could log in via Telnet, or look up people using the Finger. Furthermore, the company would be spared from having employees spend all day reading USENET news.

Blocking outgoing packey if tricky because although most sites stick to the standard port naming conventions, they are not forced the two protocol stacks were developed concurrently. In some ways, TCP/IP contributed to OSI, and vice-versa. Several important differences do exist, though, which arise from the basic requirements of TCP/IP which are:

- A common set of applications
- Dynamic routing
- Connectionless protocols at the networking level
- Universal connectivity
- Packet-Switching

The main difference between the OSI architecture and that of TCP/IP relate to the layer above the transport layer and those at the network layer. OSI has both, the session layer and the presentation layer, whereas TCP/IP combines both into an application layer. The requirement for a connectionless protocol also required TCP/IP to combine OSI's physical layer and data link layer into a network level.

Physical Layer

There is no difference between OSI and TCP/IP in respect to physical layer.

Data Link Layer

The data link layer in the OSI World makes use of the Q. 921 Lap D protocol which must support an information field length of at least 512 octets. Lap D is based on HDLC framing.

In the internet world there is no real data link layer protocol, but the subnet protocol which has quite many similarities. LANs, LLC protocol is equally used in OSI and TCP/IP networks.

Network Layer

The network layer provides routing capabilities between source and destination system.

OSI uses the *CLNS (Connection Less Network Service)* protocols ES-TS for communication of an end system to an intermediate system and TS-IS for communication between intermediate systems.

TCP divides messages in datagrams of up to 64 k length. Each datagram consists of a header and a text part. Besides some other information, the header contains the source and the destination address of the datagram. IP routes these datagrams through the network using OSPF and RIP protocols for path calculation purposes. The service provided by IP is not reliable. Datagrams may be received in the wrong order or they may even get lost in the network.

Transport Layer

Transport layer provides a reliable end-to-end connection between source and destination system on top of the network layer.

The OSI transport layer protocol (TP4) and the internet transport protocol (TCP) have many similarities but also some remarkable differences. Both protocols are built to provide a reliable connection oriented end-to-end transport service on top of an unreliable network service TP4 and TCP have a connect, transfer and a disconnect phase. The principles of doing this are also quite similar.

One difference between TP4 and TCP is that TP4 use nine different TPDU types whereas TCP known only one.

Another difference is the way both protocol react in case of a call collision.

TP4 uses a different flow control mechanism for, its message, it also provides means for quality of service measurement.

10.8 TCP/IP PROTOCOL SUIT OVERVIEW

The best place to start looking at TCP/IP is probably the name itself. TCP/IP consists of dozens of different protocols, but only a few are the “main” protocols that define the core operation of the suite. Of these key protocols, two are usually considered the most important.

The **Internet Protocol** (IP) is the primary OSI network layer protocol that provides addressing, datagram routing and other functions in an internetwork. The **Transmission Control Protocol** (TCP) is the primary transport layer protocol, and is responsible for **connection** establishment and management and reliable data transport between software processes on devices.

Due to the importance of these two protocols, their abbreviations have come to represent the entire suite: “TCP/IP”. The protocol suite as a whole requires the work of many different protocols and technologies to make a functional **network** that can properly provide users with the applications they need.

TCP/IP uses its own four-layer architecture that corresponds roughly to the OSI reference model and provides a framework for the various protocols that comprise the suite. It also includes numerous high-level **applications**, some of which are well-known by Internet users who may not realize they are part of TCP/IP, such as **HTTP** and **FTP**.

10.8.1 TCP/IP Vs OSI Model

This topic gives a brief comparison between OSI and TCP/IP protocols with a special focus on the similarities and on how the protocols from both worlds map to each other. The adoption of TCP/IP does not conflict with the OSI standards because the two protocol stacks were developed concurrently. In some ways, TCP/IP contributed to OSI, and vice-versa.

Several important differences do exist, though, which arise from the basic requirements of TCP/IP which are:

- A common set of applications
- Dynamic routing
- Connectionless protocols at the networking level.
- Packet switching.

The main differences between the OSI architecture and that of TCP/IP relate to the layers above the transport layer and those at the network layer. OSI has both, the session layer and the presentation layer, whereas TCP/IP combines both into an application layer. The requirement for a connectionless protocol also required TCP/IP to combine OSI’s physical layer and data link layer into a network level.

10.8.2 Network Layer In The Internet

At the network layer, the internet can be viewed as a collection of Subnet-works or Autonomous Systems (ASes) that are connected together. There is no real structure, but several major backbones exist. These are constructed from high-bandwidth lines and fast routers. Attached to the backbone are regional (midlevel) networks, and attached to these regional networks are the LANs at many universities, companies, and Internet Service Providers. The Figure 10.8 shows an internetwork.

The glue that holds the internet together is the network layer protocol, **IP (Internet Protocol).** Unlike most older network layer protocols, it was designed from the beginning with internetworking in mind. A good way to think of the network layer is this, as its job is to provide a best-efforts way to transport datagrams from source to destination, without regard to whether or not these machines are on the same network or whether or not there are other networks in between them.

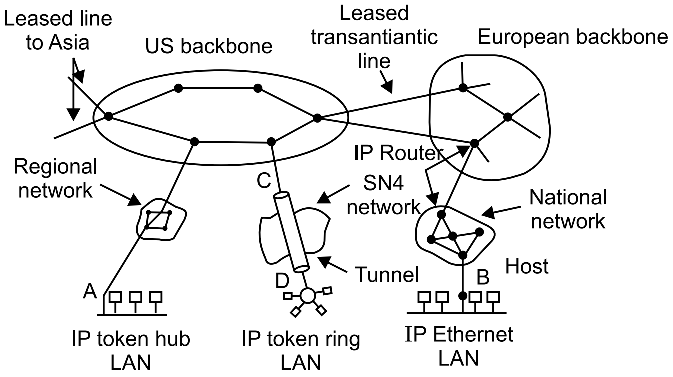


Fig. 10.8 The internet is an interconnected collection of many networks

Communication in the Internet works as follows: The transport layer takes data streams and breaks them up into datagrams. In theory, datagrams can be up to 64 Kbytes each, but in practice they are usually around 1500-bytes. Each datagram is transmitted through the Internet, possible being fragmented into smaller units as it goes. When all the pieces finally get to the destination machine, they are reassembled by the network layer into the original datagram. This datagram is then handed to the transport layer, which inserts it into the receiving process input stream.

10.8.3 Internet Protocol (IP)

First we will see the format of IP datagrams themselves. An IP datagram consists of a header part and a text part. The header has a 20-byte fixed part and a variable length optional part. The header format is shown in Figure 10.9. It is transmitted in big Indian order *i.e.*, from left to right, with the high-order bit of the version field going first. A little Indian machines software conversion is required on both transmission and reception.

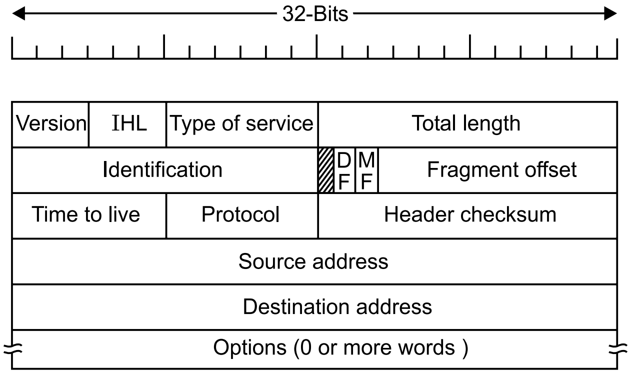


Fig. 10.9 The IP (Internet Protocol) header

Version

The version field keeps track of which version of the protocol the datagram belongs to. By including the version in each datagram, it becomes possible to have the transition between versions which would take months, or even years, with some machines running the old version and others running the new one.

IHL

Since the header length is not constant, a field in the header, IHL, is provided to tell how long the header is, in 32-bit words. The minimum value is 5, which applies when no options are present. The maximum value of this 4-bit field is 15, which limits the header to 60-bytes, and thus the options field to 40-bytes. For some options, such as one that records the route a packet has taken, 40-bytes is far too small, making the option useless.

Type of Service

The type of service field allows the host to tell the subnet what kind of service it wants. Various combinations of reliability and speed are possible. For digitized voice, fast delivery beats accurate delivery. For file transfer, error free transmission is more important than fast transmission.

The field itself contains (from left to right), a three bit precedence field, three flags, D, T and R, and 2 unused bits. The precedence field is a priority from 0 (normal) to 7 (network control packet). The three flag bits allow the host to specify what it cares most about from the set, {Delay, Throughput, Reliability}. In theory, these fields allow routers to make choices between, for example, a satellite link with high throughput and high delay or a leased line with low throughput and low delay.

In practice, current routers ignore the type of service field altogether.

Total Length

The total length includes everything in the datagram, both header and data. The maximum length is 65,535 bytes. At present, this upper limit is tolerable, but with future gigabit networks larger datagrams will be needed.

Identification

The identification field is needed to allow the destination host to determine which datagram a newly arrived fragment belongs to. All the fragments of a datagram contain the same identification value.

Next comes an unused bit and then two 1-bit fields.

D.F. Stands for Don’t Fragment and

M.F. Stands for More Fragment

Fragment Offset

The fragment offset tells where this fragment belongs in the current datagram. There is a maximum of 8192 fragments per datagram, giving a maximum datagram length of 65,536 bytes, one more than the total length field.

Time to Live

The time to live field is a counter used to limit packet lifetimes. It is suppose to count time in seconds, allowing a maximum lifetime of 255 sec. This feature prevents datagrams for wandering around forever, something that otherwise might happen if the routing tables ever become corrupted.

Protocol

The protocol field tells it which transport process to give it to. TCP is one possibility, but so are UDP and some others. The numbering of protocols is global across the entire Internet and is defined in RFC 1700.

Header Checksum

The header checksum verifies the header only. Such a checksum is useful for detecting errors generated by bad memory words inside a router, the header checksum must be recomputed at each hop, because at least one field always changes, but tricks can be used to speed up the connection. Source address and destination address:

They indicate the network number and host number.

Options

The option field was designed to provide an escape to allow subsequent versions of the protocol to include information not present in the original design, to permit experimenters to try out new ideas, and to avoid allocating header bits to information that is rarely needed. The options are variable length. The option field is padded out to a multiple of four bytes. Currently five options are defined.

Security

This option tells how secret the information is.

Strict Source Routing

This option gives the complete path from source to destination as a sequence of IP addresses.

Loose Source Routing

This option requires the packet to traverse the list of routers specified, and in the order specified, but it is allowed to pass through other routers on the way.

Record Route

This option tells the routers along the path to append their IP address to the option field.

Time stamp

This option is like the Record route option, except that in addition to recording its 32-bit IP address, each router also records a 32-bit time stamp. This option, too is mostly for debugging routing algorithms.

10.8.4 IP Addresses

An IP address is a unique, numeric identifier used to specify a particular host computer on a particular network, and is a part of a global, standardised scheme for identifying machines that are connected to the Internet.

IP addresses consist of four numbers between 0 and 255 separated by periods, which represent, both network and the host machine. No two machines have the same IP address. All IP addresses are 32-bits long and are used in the source address and destination address fields of IP packets. The formats used for IP address are shown in Figure 10.10. Those machines connected to multiple networks have a different IP address on each network.

The Class A, B, C and D formats allow for up to 126 networks with 16 million hosts each, 16382 networks with up to 64 k hosts, 2 million network, (e.g. LANs). With up to 254 hosts each, and multicast in which a datagram is directed to multiple hosts. Address beginning with 11110 are reserved for future use. Tens of thousands of networks are now connected to the Internet, and the number doubles every year. Network numbers are assigned by the NIC (Network Information Center) to avoid conflicts.

Network addresses, which are 32-bit numbers, are usually written in **dotted decimal notation**. In this format, each of the 4-bytes is written in decimal, from 0 to 255.

Example: The hexadecimal address CO 290614 is written as 192.41.6.20. The lowest IP address is 0.0.0.0 and the biggest is 255.255.255.255.

The values 0 and – 1 have special meanings, as shown in Figure 10.11. The value 0 means this network on this host. The value of – 1 is used as a broadcast address to mean all hosts on the indicated network.

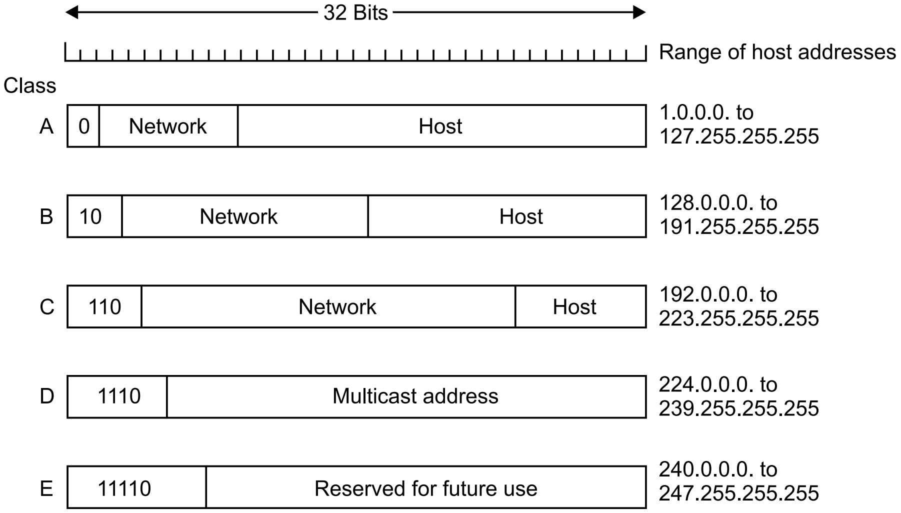


Fig. 10.10 IP Address formats

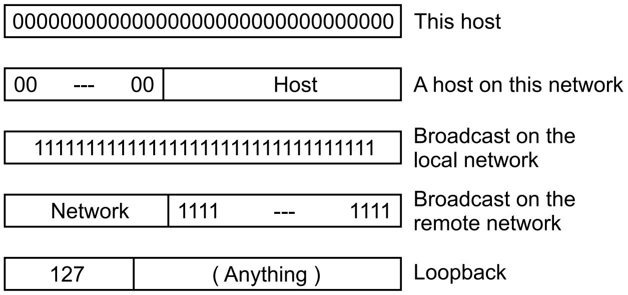


Fig. 10.11 Special IP addresses

The IP address 0.0.0.0 is used by hosts when they are being booted but is not used afterward. IP addresses with 0 as network number refer to the current network. These addresses allow machines to refer to their own network without knowing its number. The address consisting of all 1s allows broadcasting on the local network, typically a LAN. The addresses with a proper network number and all 1s in the host field allow machines to send broadcast packets to distant LANs anywhere in the Internet. Finally, all addresses of the form 127.XX.YY.ZZ are reserved for loopback testing packets sent to that address are not put onto the wire, they are processed locally and treated as incoming packets. This allows packets to be sent to the local network without the sender knowing its number. This feature is also used for debugging network software.

10.8.5 Subnetting

As we have seen, all the hosts in a network must have the same network number. This property of IP addressing can cause problems as network grow.

The solution of these problems is to allow a network to be split into several parts for internal use but still act like a single network to the outside world. In the Internet literature, these parts are called **subnets**, means the set of all routers and communication lines in a network. Hopefully it will be clear from the context which meaning is intended. In this section, the new definition will be the one used. If our growing company started up with a class B address instead of a class C address, it could start our just numbering the hosts from 1 to 254. When the second LAN arrived, it could decide, for example, to split the 16-bit host number into a 6-bit subnet number and a 10-bit host number as shown in Figure 10.12.

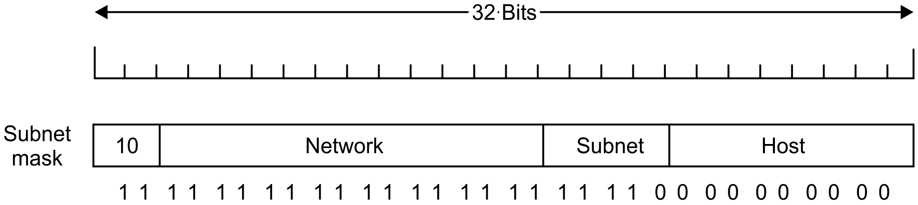


Fig. 10.12 One of the ways to subnet a class B network

Outside the network, the subnetting is not visible, so allocating a new subnet does not require contacting NIC or changing any external databases. In this example, the first subnet might use IP addresses starting at 130.50.4.1, the second subnet might start at 130.50.8.1, and so on.

To see how subnets work, it is necessary to explain how IP packets are processed at a router. Each router has a table listing some number of IP addresses. The **first kind** tells how to get to distant networks. The **second kind** tells how to get to local hosts. Associated with each table is the network interface to use to reach the destination, and certain other information.

When an IP packet arrives, its destination address is looked up in the routing table. If the packet is for a distance network, it is forwarded to the next router on the interface given in the table. If it is a local host, it is sent directly to the destination. If the network is not present, the packet is forwarded to a default router with more extensive tables. This algorithm means that each router only has to keep track of other network and local hosts, not pairs, greatly reducing the size of the routing table.

When subnetting is introduced, the routing tables are changed, adding entries of the form (this-network, subnet, 0) and (this-network, this-network, host). Thus a router on subnet K knows how to get to all the other subnets and also how to get to all the hosts on subnet K. It does not have to know the details about hosts on other subnets. In fact, all that needs to be changed is to have each router do a Boolean AND with the network's **subnet mask**, to get rid of, the host number and look up the resulting address in its table as shown in Figure subnetting thus reduces router table space by creating a three-level hierarchy.

10.8.6 Internet Control Protocol

In addition to IP, which is used for data transfer, the Internet has several control protocols used in the network layer, including ICMP, ARP, RARP, IGMP.

10.8.6.1 ICMP (Internet Control Message Protocol)

In RFC 792 ICMP is concerned with connection services. ICMP provides error control and network layer flow control, used with IP to augment error handling and control procedures.

The operation of the Internet is monitored closely by the routers. When something unexpected occurs, the event is reported by the ICMP, which is also used to test the Internet. About a dozen types of ICMP messages are defined. The most important ones are given below.

Destination unreachable

The destination unreachable message is used, when the subnet or a router cannot locate the destination or a packet with the DF bit cannot be delivered because a "small-packet" network stands in the way.

Time exceeded

The time exceeded message is sent when a packet is dropped due to its counter reaching zero. This event is a symptom that packets are looping, that there is enormous congestion, or that the timer values are being set too low.

Parameter problem

This message indicates that an illegal value has been detected in a header field. This problem indicates a bug in the sending host's IP software, or possibly in the software of a router transited.

Source quench

This message was formerly used to throttle hosts that were sending too many packets. When a host received this message, it was expected to slow down. It is rarely used any more because when congestion occurs, these packets tend to add more fuel to the fire. Congestion control in the Internet is now done largely in the transport layer.

Redirect

The redirect message is used when a router notices that a packet seems to be routed wrong. It is used by the router to tell the sending host about the probably error.

ECHO request and ECHO reply

These messages are used to see if a given destination is reachable and alive. Upon receiving the ECHO Message, the destination is expected to send an ECHO Reply message back.

Timestamp request and reply

These messages are similar, except that the arrival time of the message and the departure time of the reply are recorded in the reply. This facility is used to measure network performance.

In addition to these messages, there are four others that deal with Internet addressing, to allow hosts to discover their network numbers and to handle the case of multiple LANs sharing a single IP address.

10.8.6.2 ARP (Address Resolution Protocol)

Already we know that importance of IP addressing. In simple terms, it makes addressing on the Internet uniform. However, having only the IP address of a node is not good enough. There must be a process for obtaining the physical address of a computer based on its IP address, in order to be able to finally actually transmit the frame/datagram over the network to which the node belongs. This process is called **address resolution.** This is required because at the hardware level, computers identify each other based on the physical addresses hard-coded on their Network Interface Cards (NICs). They neither know the relationship between the IP address prefix and a physical network, nor the relationship between on IP address suffix and a particular computer.

Networking hardware demands that a datagram contain the physical address of the intended recipient. It has no clue to the IP addresses. To solve the problem, the Address Resolution Protocol (ARP) was developed. ARP takes the IP address of a host as input and gives its corresponding physical address as the output which is shown in Figure 10.13.

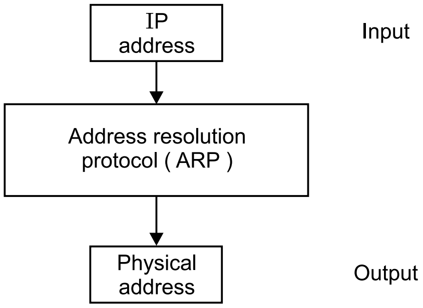


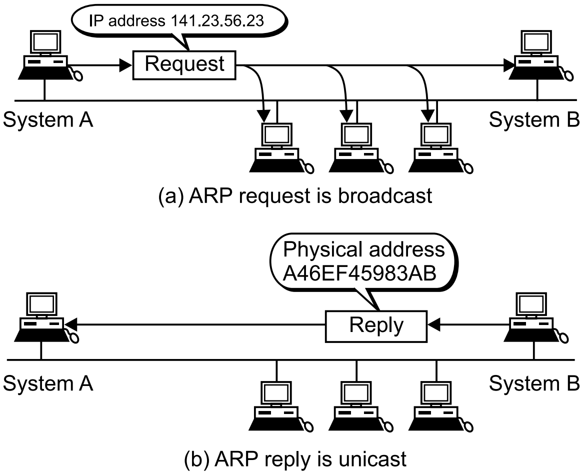
Fig. 10.13 Address resolution protocol (ARP)

Any time a host, or a router, needs to find the MAC address of another host or router on its network, it sends an ARP query packet. The packet includes the physical and IP addresses of the sender and the IP address of the receiver. Because the sender does not know the physical address of the receiver, the query is broadcast over the network as shown in Figure 10.14.

Every host or router on the network receives and processes the ARP query packet, but only the intended recipient recognizes its IP address and sends back on ARP response packet. The response packet contains the recipient's IP and physical addresses.

The packet is unicast directly to the inquirer using the physical address received in the query packet.

In the Figure 10.14 (a), the System on the left (A) has a packet that needs to be delivered to another System (B) with IP address 141.23.56.23. System A needs to pass the packet to its data link layer for the actual delivery, but it does not know the physical address of the recipient. It uses the services of ARP to send a broadcast request packet to ask for the physical address of a system with an IP address of 141.23.56.23.



(a) ARP request is broadcast

(b) ARP reply is unicast

Fig. 10.14 ARP operation

This packet is received by every system on the physical network, but only System B will answer it, as shown in Figure 10.14 (b). System B sends on ARP reply packet that includes its physical address. Now System A can send all the packets it has for this destination, using the physical address it received.

10.8.6.3 RARP (Reverse Address Resolution Protocol)

There is one more protocol in the ARP suite of protocols. The **Reverse Address Resolution Protocol (RARP)** is used to obtain the IP address of a host based on its physical address. That is, it performs a job that is exactly opposite to that of the ARP. An obvious question would be : is this really needed? After all, a host should have the IP address stored on its hard disk! However, there are situations when this is not the case. **Firstly**, a host may not have a hard disk at all (*e.g.,* a diskless workstation). **Secondly**, when a new host is being connected to a network for the very first time, it does not know its IP address. Finally, a computer may be discarded and replaced by another computer, but the same network card could be reused. In all these situations, the computer does not know it own IP address.

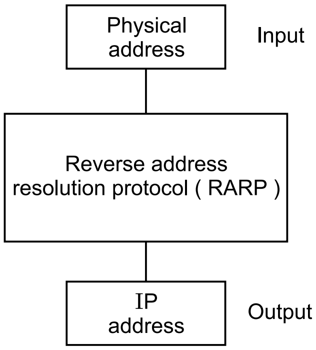
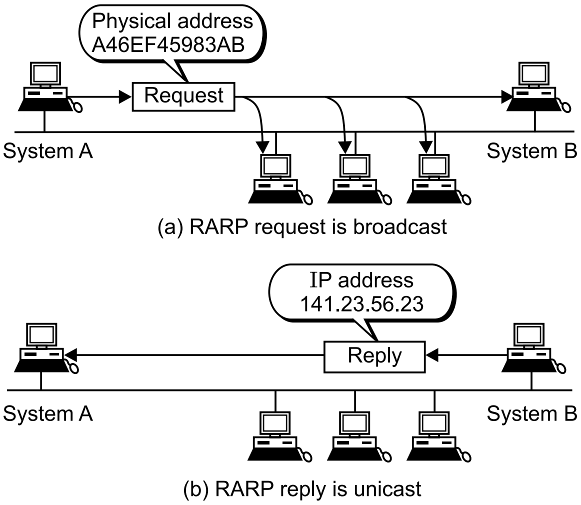


Fig. 10.15 Reverse Address Resolution Protocol (RARP)

RARP works in very similar way to ARP, but in the exactly opposite direction as shown in Figure 10.15.

In RARP, the host interested in knowing its IP address broadcasts on RARP query datagram. This datagram contains its physical address. Every other computer on the network receives the datagram. All the computers except a centralized computer (the server computer) ignore this datagram. However, the server recognizes this special kind of datagram and sends the broadcasting computer its IP address. The server contains a list of the physical addresses and their corresponding IP addresses for all diskless workstations which is shown in Figure 10.16.



(a) RARP request is broadcast

(b) RARP reply is unicast

Fig. 10.16 RARP operation

In the Figure 10.16 (a), the system on the left (A) has a packet that needs to be delivered to another System (B) with physical address A46EF45983AB. System A needs to pass the packet to its data link layer for the actual delivery, but it does not know the IP address of the recipient. It uses the services of RARP to send a broadcast request packet to ask for the IP address of a system with the physical address 0FA46EF45983AB.

This packet is received by every system on the physical network, but only System B will answer it, as shown in Figure 10.16 (b). System B sends an RARP reply packet that include its IP address. Now System A can send all the packets it has for this destination using the IP address it received.

10.8.6.4 IGMP (Internet Group Message Protocol)

The Internet Group Management Protocol (IGMP) is one of the necessary protocol involved in multicasting.

IGMP is not a multicasting routing protocol. It is a protocol that manages **group membership**. In any network there are one or more **multicast routers** that distribute multicast packets to hosts or other routers. IGMP gives the multicast routers information about the membership status of hosts or routers connected to the network.

A multicast router may receive thousands of multicast packets every day for different groups. If a router has no knowledge about the membership status of the hosts, it must broadcast all these packets. This creates a lot of traffic and consumes bandwidth. A better solution is to keep a list of groups in the network for which there is at least one loyal member. IGMP helps the multicast router create and update this list.

IGMP operates locally. A multicast router connected to a network has a list of multicast addresses of the groups for which the router distributes packets to groups with at least one loyal member in that network as shown in Figure 10.17.

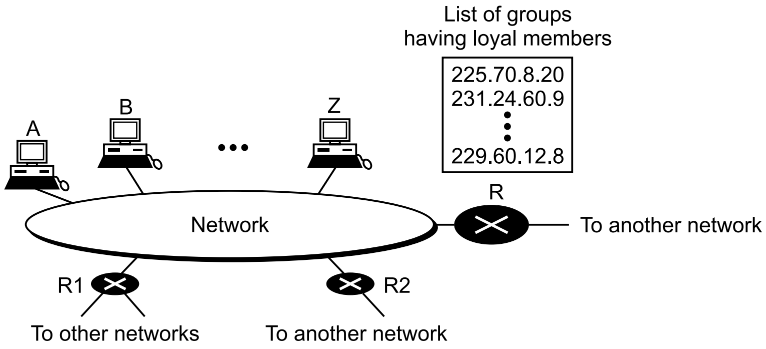


Fig. 10.17 IGMP operation

For each group, there is one router which has the duty of distributing the multicast packets destined for that group. This means that if there are three multicast routers connected to a network, their lists of groupids are mutually exclusive. For example, in the Figure 10.17 only router R distributes packets with the multicast address of 225.70.8.20.

A host or multicast router can have membership in a group. When a host has membership, it means that one of its processes receives multicast packets from some group. When a router has membership, it means that a network connected to one of its other interfaces receives these multicast packets. We say that the host or the router has an interest in the group. In both cases, the host and the router keep a list of groupids and relay their interest to the distributing router.

For example, in the given figure, router R is the distributing router. There are two other multicast routers (R_1 and R_2) which, depending on the group list maintained by router R, could be the recipients of router R in this network. Routers R_1 and R_2 may be distributors for some of these groups in other networks, but not on this network.

QUESTIONARIES

1. Discuss various Internetworking devices and their uses in different layers.

2. Explain how internetworking with concatenated virtual circuits takes place.

3. What are advantages of Internetworking with concatenated virtual circuits.

4. Explain the concept of connectionless Internet.

5. What is the purpose of Tunneling and how it takes place.

6. State various causes of Fragmentation.

7. Explain different types of Fragmentation along with their advantages and disadvantages.

8. State the possible problem that can occur during Fragmentation.

9. Explain the concept of firewalls.

10. Explain the functions of application gateway and packet filters in the implementation of Firewall.

11. Discuss network layer in the Internet.

12. With the suitable example, explain the concept of subnet mask.

13. Explain ICMP along with its message types.

14. How ARP resolves IP addresses into MAC addresses.

15. Draw and explain IP header format.

16. Explain special IP addresses a class ‘B’ network has subnet mask of 255.255.240.0. What is the maximum number of host per subnetwork?

17. Write short note on:

- ICMP
- ARP
- IP addresses and classification
- RARP
- IP features
- IGMP

Chapter 11

TRANSPORT LAYER

INTRODUCTION

The transport layer is not just another layer. It is the heart of the whole protocol hierarchy. Its task is to provide reliable, cost-effective data transport from the source machine to the destination machine, independent of the physical network or networks currently in use. Without the transport layer, the whole concept of layered protocols would make little sense. In this chapter we will study about transport layer.

11.1 SERVICE PRIMITIVES

The **transport service primitives** allow transport users (*e.g.*, application programs) to access the transport service. Each transport service has its own access primitives. In this section, we will first examine a simple (hypothetical) transport service and then look at a real example.

Then transport service is similar to the network service, but there are also some important differences. The main difference is that the network service is intended to model the service offered by real networks, wants and all. Real networks can lose packets, so the network service on generally unreliable.

The (connection-oriented) transport service, in contrast, is reliable of course, real networks are not error-free, but that is precisely the purpose of the transport layer-to provide a reliable service on top of an unreliable network.

As an example, consider two processes connected by pipes in **UNIX**. They assume the connection between them is perfect. They do not want to know about acknowledgements, lost packets, congestion, or anything like that. What they want is a 100 per cent reliable connection. Process A puts data into one end of the pipe, and process B takes it out of the other. This is what the connection-oriented transport service is all about-hiding the imperfections of the network service so that user processes can just assume the existence of an error-free bit stream.

As an aside, the transport layer can also provide unreliable (datagram) service. A second difference between the network service and transport service is whom the services are intended for. The network service is used only by the transport entities. Few users write their own transport entities, and thus few users or programs ever see the bare network service. The transport service must be convenient and easy to use.

To get on idea of what a transport service might be like, consider the **Five** primitives which are given below:

Table 11.1 The primitives for a simple transport service

S/R	*Primitivies*	*TPDU sent*	*Meaning*
1	LISTEN	(None)	Block until some process tries to connect.
2	CONNECT	CONNECTION REQ.	Actively attempt to establish connection.
3	SEND	DATA	Send information.
4	RECEIVE	(None)	Block until a DATA TPDU arrives.
5	DISCONNECT	DISCONNECT REQ.	This side wants to release the connection.

To see how these primitives might be used, consider an application with a server and a number of remote clients. To start with, the server executes a **LISTEN** primitive, typically by calling a library procedure that makes a system call to block the server until a client turns up. When a client wants to talk to the server, it executes a **CONNECT** primitives. The transport entity carries out this primitive by blocking the caller and sending a packet to the server. Encapsulated in the payload of this packet is a transport layer message for the server's transport entity.

A quick note on terminology is now in order. For lack of a better term, we will reluctantly use the somewhat ungainly acronym TPDU (**Transport Protocol Data Unit**) for messages sent from transport entity to transport entity. Thus TPDUs are contained in packets. In turn, packets are contained in frames. When a frame arrives. The data link layer processes the frame header and passes the contents of the frame payload field up to the network entity. The network entity process the packet header and passes the contents of the packet payload up to the transport entity. This nesting is given in Figure 11.1.

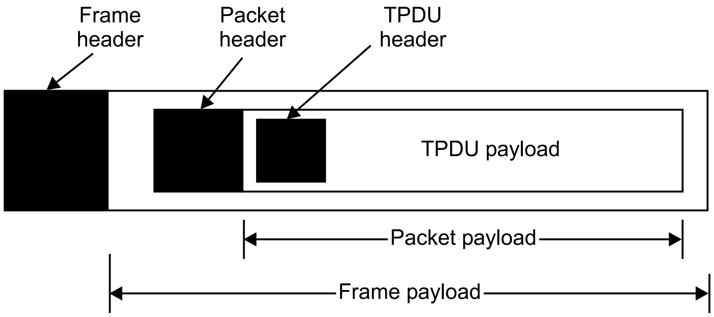


Fig. 11.1 Neting of TPDU's, packets, and frames

Getting back to our client-server example, the client's **CONNECT** call causes a CONNECTION REQUEST TPDU to be sent to the server. When it arrives, the transport entity checks to see that the server is blocked on a LISTEN. It then unblocks the server and sends a CONNECTION ACCEPTED TPDU back to the client. When the TPDU arrives, the client is unblocked and the connection is established.

Data can now be exchanged using the **SEND** and **RECEIVE** primitives. In the simplest form, either party can do a (blocking) RECEIVE to wait for the other party to do a SEND. When the TPDU arrives, the receiver is unblocked. It can then process the TPDU and send a reply.

Not that at the network layer, even a simple unidirectional data exchange is more complicated, than at the transport layer. Every data packet sent will also be acknowledge. The packets bearing control TPDUs are also acknowledged, implicitly and explicity. The acknowledgements are managed by the transport entities using the network layer protocol and are not visible to the transport users. Similarly, the transport entities will need to worry about timers and retransmissions. None of this machinery is seen by the transport users. To the transport users, a connection is a reliable bit pipe: one user stuffs bits in and they magically appear at the other end. This ability to hide complexity is the reason that layered protocols are such a powerful tool.

When a connection is no longer needed, it must be released to free up table space within the two transport entities. Disconnection has two variants: **asymmetric** and **symmetric**. In the **asymmetric** variant, either transport user can issue a **DISCONNECT** primitive, which results in a DISCONNECT TPDU being sent to the remote transport entity. Upon arrival, the connection is released.

In the **symmetric** variant, each direction is closed separately, independently of the other one. When one side does not DISCONNECT, that means it has no more data to send, but it is still willing to accept data from its partner. In this model, a connection is released when both sides have done a DISCONNECT.

Let us now briefly inspect another set of transport primitives, the socket primitives used in **Berkeley UNIX** and **TCP** and these are given in the following table:

Table 11.2 The socket primitives for TCP

Primitive	*Meaning*
SOCKET	Create a new communication end point
BIND	Attach a local address to a socket
LISTEN	Announce willingness to accept connections; give queue size
ACCEPT	Block the caller until a connection attempt arrives
CONNECT	Actively attempt to establish a connection
SEND	Send some data over the connection
RECEIVE	Receive some data from the connection
CLOSE	Release the connection

11.2 ELEMENTS OF TRANSPORT PROTOCOLS

There are significant differences between the data link layer and transport layer.

- At the datalink layer, two routers communicate directly via a physical channel, whereas at the transport this physical channel is replaced by the entire subnet.
- Another difference between the data link layer and the transport layer is the potential existence of storage capacity in the subnet.
- A final difference between the datalink and transport layers is one of amount rather than of kind. Buffering and flow control are needed in both layers, but the presence of a large and dynamically varying number of connections in the transport layer many require a different approach than we used in the data link layer.

In the following sections, we will study some important issues.

Addressing

When an application process wishes to setup a connection to a remote application process, it must specify which one to define transport addresses to which processes can listen for connection requests. In the Internet, these end points are (IP address, local port) pairs. In ATM networks, they are AAL-SAPS. We will use the neutral term TSAP (Transport Service Access Point). The analogous end points in the network layer are then called NSAPs. IP addresses are examples of NSAPs.

A possible connection scenario for a transport connection over a connection-oriented network layer is as follows:

1. A time-of-day server process on host 2 attaches itself to TSAP 122 to wait for an incoming call which is shown in Figure 11.2. How process attaches itself to a TSAP is outside the networking model and depends entirely on the local operating system. A call such as our LISTEN might be used, for example.
2. An application process on host 1 wants to find out the time-or-day, so it issues a CONNECT request specifying TSAP 6 as the source and TSAP 122 as the destination.
3. The transport entity on host 1 selects a network address on its machine and sets up a network connection between them. Using this network connection, host is transport entity can talk to the transport entity on host 2.
4. The first thing the transport entity on 1 says to its peer on 2 is: “Good Morning, I would like to establish a transport connection between my TSAP 6 and your TSAP 122. What do you say?”
5. The transport entity on 2 then asks the time-of-day server at TSAP 122 if it is willing to accept a new connection. If it agrees, the transport connection is established.

Figure 11.2 illustrates the relationship between the NSAP, TSAP, network connection, and transport connection for a connection-oriented subnet (*e.g.*, ATM).

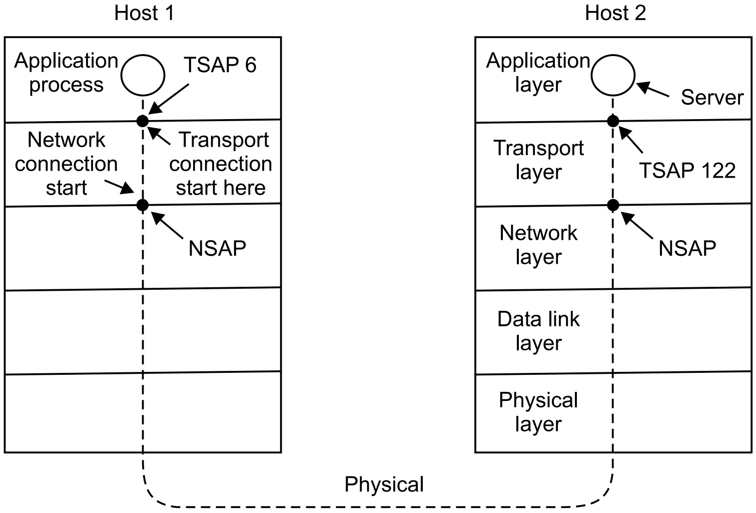


Fig. 11.2 TSAPs, NSAPs, and Connections

Establishing a Connections

Establishing a connection sounds easy, but it is actually surprisingly tricky. At first glance, it would seem sufficient for one transport entity to just send a CONNECTION REQUEST TPDU to the destination and wait for a CONNECTION ACCEPTED reply. The problem occurs when the network can lose, store, and duplicate packets.

Consider one example, A user establishes a connection with a back, sends messages telling the bank to transfer a large amount of money to the account of a not-entirely-trust worthy person, and then releases the connection. Unfortunately, each packet in the scenario is duplicated and stored in the subnet.

The crux of the problem is the existence of delayed duplicates. It can be attacked in various ways, none of them very satisfactory. One way is to use throwaway transport addresses.

Another possibility is to give each connection, a connection identifier (*i.e.*, a sequence number incremented for each connection established), chosen by the initiating party, and put in each TPDU, including the one requesting the connection.

Releasing a Connection

Releasing a connection is easier than establishing one. Nevertheless, there are more pitfalls than one might expect. As we mentioned earlier, there are two styles of terminating a connection.

- Asymmetric release
- Symmetric release
- Asymmetric release is the way the telephone, system works: When one party hangs up, the connection is broken.

- Symmetric release treats the connection as two separate unidirectional connections and requires each one to be released separately.

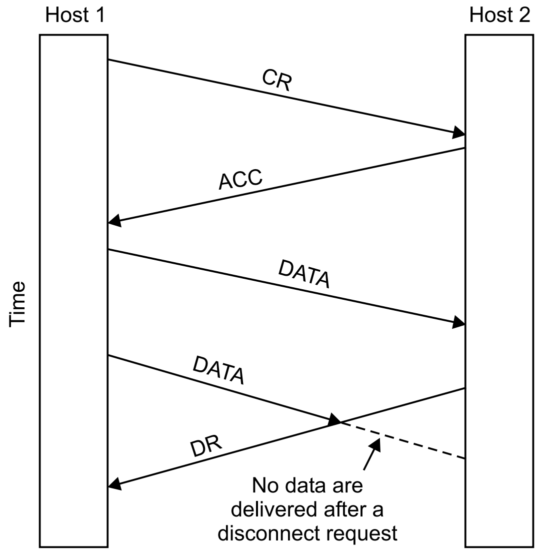


Fig. 11.3 Abrupt disconnection with loss of data

Asymmetric release is abrupt and may result in data loss. Consider the scenario, which is shown in Fig. 11.3. After the connection is established, host 1 sends a TPDU that arrives properly at host 2 then host 1 sends another TPDU. Unfortunately, host 2 issues a DISCONNECT before the second TPDU arrives. The result is that the connection is released and data are lost.

Clearly, a more sophisticated release protocol is required to avoid data loss. One way is to use symmetric release, in which each direction is released independently of the other one. Here, a host can continue to release data even after it has sent a DISCONNECT TPDU.

Symmetric release does the job when each process has a fixed amount of data to send and clearly knows when it has sent it. In other situations, determining that all the work has been done and the connection should be terminated is not so obvious. One can envision a protocol in which host ; says : “I am done, Are you done too?” if host 2 responds : “I am done too. Goodbye .” the connection can be safely released.

Flow Control and Buffering

We have observed a connection establishment and connection release in some detail and now. We have to look at how connections are managed while they are in use. One of the key issues has come up before: flow, control. In some ways the flow control problem in the transport layer is the same as in the data link layer, but in other ways it is different. The basic similarity is that in both layer, a sliding window, or other scheme is needed on each connection to keep a fast transmitter from over running a slow receiver. The main different is that a router usually has relatively few lines whereas a host may have numerous connections.

In the data link protocol, frames were buffered at both the sending router and at the receiving router.

In the data link layer, the sending side must buffer outgoing frames because they might have to be retransmitted. If the subnet provides datagram service, the sending transport entity must also buffer and for the same reason. If the receiver knows that the sender buffer all TPDUs unit they are acknowledged, the receiver may or may not dedicate specific buffers to specific connections, as it sees fit.

Multiplexing

Multiplexing several conversations on to connections virtual circuits, and physical links plays a role in several layers of the network architecture. In the transport layer the need for multiplexing can arise in a number of ways.

Upward Multiplexing

The consequence of a price structure that heavily penalizes installations for having many virtual circuits open for long periods of time is to make multiplexing of different transport connections onto the same network connection attractive. This form of multiplexing, called as upward multiplexing, as shown in Figure. 11.4.

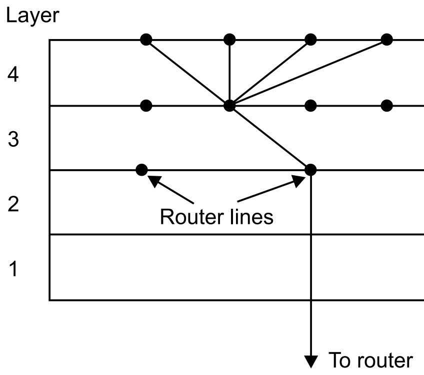


Fig. 11.4 Upward multiplexing

In this figure, four distinct transport connections all use the same network connection (*e.g.*, ATM virtual circuit) to the remote host. When connect time forms the major component of the carrier's bill, it is upto the transport layer to group transport connections according to their destination and map each group on to the minimum number of network connection. If too many transport connections are mapped onto one network connection, the performance will be poor. Because the window will usually be full, and users will have to wait their turn to send one message and the service will be expensive.

When upward multiplexing is used with ATM, we have the ironic situation of having to identify the connection using a field in the transport header, even though ATM provides more than 4000 virtual circuit numbers/virtual path expressly for that purpose.

Downward Multiplexing

Multiplexing can also be useful in the transport layer related to carrier technical decisions rather than carrier pricing decisions. And one possible solution is to have the transport layer open multiple network connections and distribute the traffic among them on a round robin basis, as shown in Figure 11.5. This modus operande is called downward multiplexing.

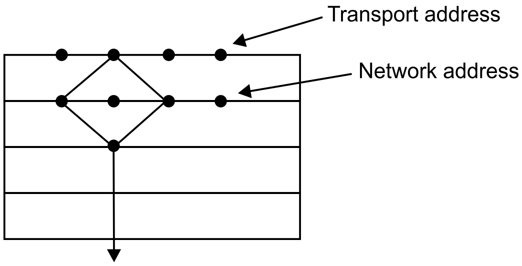


Fig. 11.5 Downward multiplexing

With K network connections open, the effective bandwidth is increased by a factor of K. With 4095 virtual circuits, 128-byte packets, and an 8-bit sequence number. It is theoretically possible to achieve data rates in excess of 7.6 Gbps.

Crash Recovery

Recovery becomes an issue, if hosts and routers are subject to crashes. If the transport entity is entirely within the hosts, recovery from network and router crashes is straight forward.

When the network layer provides datagram service, the transport entities expect lost TPDUs all the time. Also when the network layer provides connection-oriented service, then loss of a virtual circuit is handled by establishing a new one and then probing the remote transport entity to ask it which TPDUs it has received and which ones it has not received. The latter ones can be retransmitted.

A more troublesome problem is how to recover from host crashes. For illustrating this problem, let us assume that one host, the client, is sending a long file to another host, the file server, using a simple stop-and-wait protocol. The transport layer on the server simply passes the incoming TPDUs to the transport user, one by one. Part way through the transmission, the server crashes. When it comes back up, its tables are reinitialized, so it no longer knows precisely where it was.

The server can be programmed in one of two ways:

1. Acknowledge first
2. Write first

Each client can be in one of two states.

- One TPDU outstanding, S1
- No TPDUs outstanding, S0

The client can be programmed in one of four ways:

1. Always retransmit the last TPDU
2. Never retransmit the last TPDU
3. Retransmit only in State S0
4. Retransmit only in State S1

This gives eight combinations, but as we shall see, for each combination there is some set of events that makes the protocol fail. There are three events possible at the server.

1. Sending an acknowledgement (A)
2. Writing to the output process (W)
3. Crashing (C)

These three events can occur in six different orderings AC(W), AWC, C(AW), C(WA), WAC and WC(A), where the parentheses are used to indicate that neither A nor W may follow C (*i.e.*, once it has crashed, it has crashed).

11.3 TCP

TCP (**Transmission Control Protocol**) is connection-oriented protocol TCP was specifically designed to provide a reliable end-to-end byte stream over an unreliable internetwork. Also TCP was designed to dynamically adapt to properties of the internetwork and to be robust in the face of many kinds of failures.

TCP was formally defined in RFC 793 ; TCP was detailed defined in RFC 1122. Extensions are given in RFC 1323.

Each machine supporting TCP has a TCP transport entity, either a user process or post of the kernel that manages TCP stream and interfaces to the IP layer. A TCP entity accepts user data streams from local processes, breaks them up into pieces not exceeding 64 K bytes, and sends each piece as a separate IP datagram. When IP datagrams containing TCP data arrive at a machine, they are given to the TCP entity, which reconstructs the original byte streams. Sometimes we just "TCP" to mean the TCP transport entity (a piece of software) or the TCP protocol (a set of rules). **TCP** must furnish the reliability that most users want and that IP does not provide.

The TCP Service Model

TCP service is obtained by having both the sender and receiver create end points, called **sockets**. Each socket has a socket number (address) consisting of the IP address of the host and a 16-bit number local to that host, called a **port**. A port is the TCP name for a TSAP. To obtain TCP service, a connection must be explicitly established between a socket on the sending machine and a socket on the receiving machine. The socket calls are listed in Table 11.2.

A socket may be used for multiple connections at the same time and connections are identified by the socket identifiers at both ends, *i.e.*, (Socket 1, Socket 2). No virtual circuit numbers or other identifiers are used.

Port numbers below 256 are called **well-known ports** and are reserved for standard services. To establish a remote login session using TELNET, port **23** is used. The list of well-known ports is given in RFC 1700.

All TCP connections are full-duplex and point to point. TCP does not support multicasting or broadcasting. A TCP connection is a byte stream, not a message stream. Message boundaries are not preserved **end-to-end**. There is no way for the receiver to detect the unit(s) in which the data were written.

When an application passes data to TCP, TCP may send it immediately or buffer it, at its discretion. However, sometimes, the application really wants the data to be sent immediately. For forcing data out, applications can use the PUSH flag, which tells TCP not to delay the transmission. TCP is free to collect all the PUSHED than into a single IP datagram, with no separation between the various pieces.

One last feature of the TCP service that is worth mentioning here is **Urgent data**. When an interactive user hits the DEL or CTRL-C key to break off a remote, computation that has already begun, the sending application puts some control information in the data stream and gives it to TCP along with the **URGENT** flag. This event causes TCP to stop accumulating data and transmit everything, it has for that connection immediately.

When the urgent data are received at the destination, the receiving application is interrupted.

TCP Protocol

Here we will focus on a general overview of the TCP protocol. Every byte on a TCP connection has its own 32-bit sequence number. For a host blasing away at full speed on a 10 Mbps LAN. The sequence numbers are used both for acknowledgement and for the window mechanism, which use separate 32-bit header fields.

The sending and receiving TCP entities exchange data in the form of segments. A **segment** consists of a fixed 20-byte header followed by zero or more data types. The TCP software decides how big segments should be. It can accumulate data for several writes into one segment or split data from one write over multiple segments. Two limits restrict the segment size.

1. Each segment, including the TCP header, must fit in the 65,535 byte IP payload.
2. Each network has a maximum transfer unit or **MTU**, and each segment must fit in the MTU.

A segment that is two large for a network that is must transit can be broken up into multiple segments by a router. Each new segment gets its own TCP and IP headers, so fragmentation by routers increase the total overhead.

The basic protocol used by TCP entities is the sliding window protocol. When a sender transmits a segment, it also starts a timer. When the segment arrives at the destination, the receiving TCP entity sends back a segment bearing on acknowledgement number equal to the next sequence number it expects to receive. If the sender's times goes off before the acknowledgement is received, the sender transmits the segment again.

Although this protocol sounds simple. TCP must be prepared for solving problems in an efficient way.

The TCP Segment Header

Figure 11.6 shows the layout of a TCP segment.

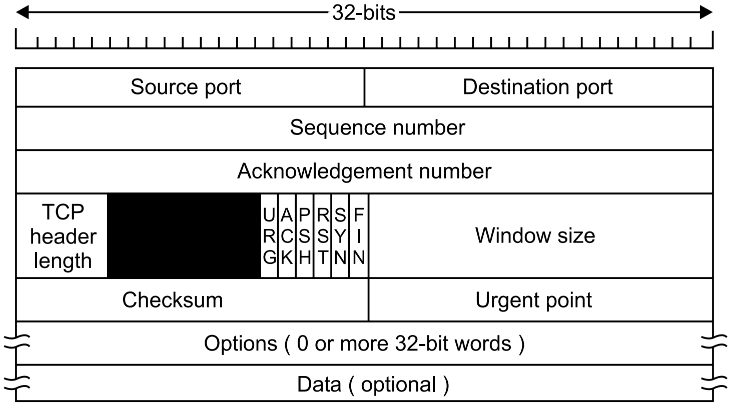


Fig. 11.6 The TCP header

- The **Source port** and **Destination port** fields identify the local end points of the connection. Each host may decide for itself how to allocate its own ports starting at 256.
- The **Sequence number** and **Acknowledgement number** field perform their usual functions.
- The **TCP header length** tells how many 32-bit words are contained in the TCP header. Technically this field really indicates the start of the data within the segment, measured in 32-bit words, but that number is just the header length in words.

  Next comes a 6-bit field that is not used the fact that this field has survived intact for over a decade is testimony to how well thought out TCP is. Now come six 1-bit flags. URG is set to 1 if the **Urgent pointer** is in use.
- The **Urgent pointer** is used to indicate a byte offset from the current sequence number at which urgent data are to be found.
- The **Ack bit** is set to 1 to indicate that the acknowledgement number is valid, if Ack is 0, acknowledgement number field is ignored.

* The **PSH bit** indicates PUSHed data.

* The **RST bit** is used to reset a connection that has become confused due to a host crash or some other reason. It is also used to reject an invalid segment or refuse an attempt to open a connection.

* The **SYN bit** is used to establish connections.

* The **FIN bit** is used to release a connection.

The **Flow control** in TCP is handled using a variable-size sliding window.

* A **checksum** is also provided for extreme reliability. It checksums the header, the data, and the conceptual pseudoheader.

* The **options field** was designed to provide a way to add extra facilities not covered by the regular header.

* After that, it can acknowledge all the buffered **data**, thus reducing the amount of data retransmitted.

TCP Connection Management

Connections are established in TCP using the three-way handshake. To establish a connection at the server side passively waits for an incoming connection by executing the LISTEN and ACCEPT primitives, either specifying a specific source or nobody in particular.

The other side, *i.e.*, client side, executes a CONNECT primitive, specifying the IP address and port to which it wants to connect, the maximum TCP segment size it is willing to accept, and optionally some user data (*e.g.*, a password).

The steps required to establish and release connections with the 11 states are as follows:

- **CLOSED:** Each connection starts in the CLOSED state. Here no connection is active or pending.
- **LISTEN:** The server is waiting for an incoming call.
- **SYN RCVD:** A connection request has arrived, wait for ACK.
- **SYN SENT:** The application has started to open a connection.
- **ESTABLISHED:** The normal data transfer state.
- **FIN WAIT 1:** The application has said to be finished.
- **FIN WAIT 2:** The other side has agreed to release.
- **TIMED WAIT:** Wait for all packets to die off.
- **CLOSING:** Both sides have tried to close simultaneously.
- **CLOSE WAIT:** The other side has initiated a release.
- **LAST ACK:** Wait for all packets to die off.

TCP Transmission Policy

Window management in TCP is not directly tied to acknowledgements as it is in most data link protocols.

Example: Suppose the receiver has a 4096-byte buffer as shown in Figure 11.7 If the sender transmits a 2048-byte segment that is correctly received, the receiver will acknowledge the segment. However, since it now has only 2048 of buffer space, it will advertise a window of 2048 starting at the next byte expected.

Now the sender transmits another 2048-bytes which are acknowledged, but the advertised window is 0. The sender must stop until the application process on the receiving host has removed some data from the buffer, at which time TCP can advertise a longer window.

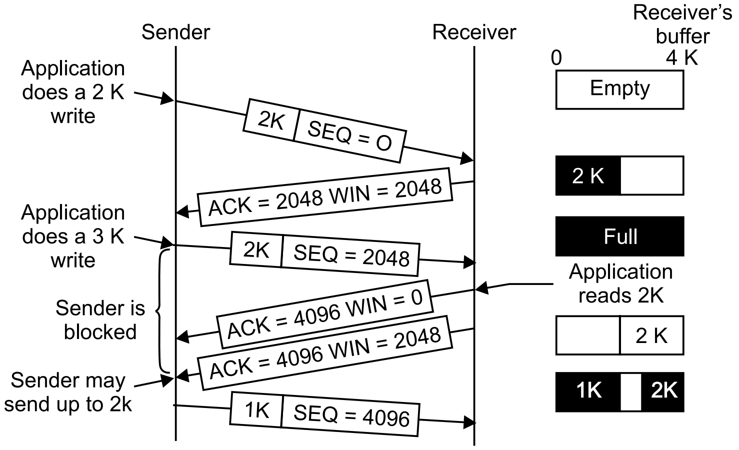


Fig. 11.7 Window management in TCP

When the window is 0, the sender may not normally send segments, with two exception. (1) urgent data may be sent, (2) the sender may send a 1-byte segment to make the receiver reannounce the next byte expected and window size.

Senders are not required to transmit data as soon as they come in from the application. Nagle's algorithm is widely used to TCP implementations, but there are times when it is better to disable it.

Another problem that can run TCP performance is the silly window syndrome (Clark, 1982) this problem occurs when data are passed to the sending TCP entity in large blocks, but an interactive application on the receiving side reads data 1-byte at a time. **Clark's Solution** is to prevent the receiver from sending a window update for 1-byte. Instead it is forced to wait until it has a decent amount of space available and advertise that instead. Specifically, the receive should not send a window update until it can handle the maximum segment size it advertised when the connection was established, or its buffer it half empty, whichever is smaller.

TCP Congestion Control

When the load offered to any network is more than it can handle, congestion builds up. The Internet is no exception. Although the network layer also tries to manage congestion, most of the heavy lifting is done by **TCP** because the real solution to congestion is to slow down the data rate.

Theoretically, congestion can be dealt with by employing a principle borrowed from Physics: the law of conservation of packets. The idea is not to inject a new packet into the network until an old one leaves. TCP attempts to achieve this goal by dynamically manipulating the window size.

The first step in managing congestion is detecting it. In the old days, detecting congestion was difficult, but now-a-days, packet loss due to transmission errors is relatively rare because most long-haul trunks are fiber (although wireless networks are a different story). Consequently, most transmission timeouts on the Internet are due to congestion. All the Internet TCP algorithms assume that timeouts are caused by congestion and monitor timeouts for signs of trouble the way miners watch their canaries.

How we can see, the Internet congestion control algorithm. It uses a third parameter, the **threshold**, initially 64 K, in addition to the receiver and congestion windows. When a timeout occurs, the threshold is set to half of the current congestion window, and the congestion window is reset to one maximum segment. Slow start is then used to determine what the network can handle, except that exponential growth stops when the threshold is hit. From that point on, successful transmissions grow the congestion window linearly instead of one per segment. In effect this algorithm is guessing that it is probably acceptable to cut the congestion window in half, and then it gradually works its way up from there.

Work on improving the congestion control mechanism is continuing.

TCP Timer Management

TCP uses multiple timers to do its work. The most important of these is the **retransmission timer**. When a segment is sent, a retransmission timer is started. If the segment is acknowledged before the timer expires, the timer is stopped. If, on the other hand, the timer goes off before the acknowledgement comes in, the segment is retransmitted. (and the timer started again).

The timeout interval can be calculated by

$$\boxed{Timeout = RTT + 4 * \mathrm{D}}$$

where RTT = Round trip time (msec)

4 = Jacobson Factor (really 2), 4 gives better performance

$$\mathrm{D} = \alpha \mathrm{D} + (1 - \alpha)\left|\mathrm{RTT} - \mathrm{M}\right|$$

where

D = Smoothed variable deviation

α = Typically 7/8; (Smoothing Factor)

M = Measurement of how long the **ack** took acknowledgement

A second timer is the **persistence timer**. It is designed to prevent the following deadlocks. The receiver sends an acknowledgement with a window size of 0, telling the sender to wait. Later, the receiver updates the window, but the packet with the update is lost. Now both the sender and receiver are waiting for each other to do something. If it is still zero, the persistence timer is set again and the cycle repeats. If it is non-zero, data can now be sent.

A third time that some implementations use is the **keepalive timer**. When a connection has been idle for a long time, the keepalive timer may go off to cause oneside to check if the other side is still there. If it fails to respond, the connection is terminated. This feature is controversial because it adds overhead and may terminate an otherwise healthy connection due to a transient network partition.

The last timer used on each TCP connection is the one used in the **TIMED WAIT** state while closing. It runs for twice the maximum packet lifetime to make sure that when a connection is closed, all packets created by it have died off.

11.4 UDP

The Internet protocol suite also supports a connectionless transport protocol, UDP **(User Data Protocol).** UDP provides a way for applications to send encapsulated raw IP datagrams and send them without having to establish a connection. Many client server applications that have one request and one response use **UDP** rather than go to the trouble of establishing and later releasing a connection.

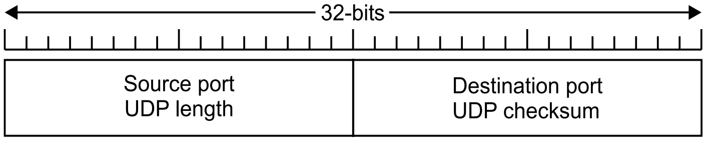


Fig. 11.8 The UDP header

A UDP segment consists of an 8-byte header followed by the data. The header is shown in Figure. 11.8. The two ports serve the some function as they do in TCP: To identify the end points within the source and destination machines. The UDP length fields includes the 8-byte header and the data. The UDP checksum includes the same format pseudoheader shown in Figure 11.9 the UDP header,

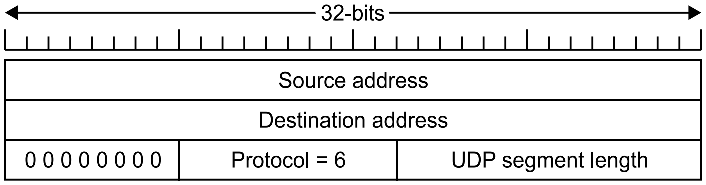


Fig. 11.9 The Pseudoheader included in the UDP checksum

and UDP data, padded out to an even number of bytes if needed. It is optional and stored as 0 if not computed (a true computed 0 is stored as all 1s, which is the same in 1's complement). Turning it off is foolishness unless the quality of the data does not matter. (*e.g.*, digitized speech).

UDP Provides Two Services

- UDP uses a port address to achieve packet delivery. Port address is simply a pointer to the process, not a connection identifier, as it is with TCP/IP. Port address helps to distinguish different user requests.
- A checksum capability to verify that the data that has arrived is intact.

QUESTIONNARIES

1. Explain the various services provided by transport layer to upper layers.
2. Discuss quality of service of transport layer with various services parameters.
3. Explain transport layer service primitives.
4. Explain the addressing scheme used by transport layer.
5. How the execution of several internet application takes place over a single IP address?
6. Explain TCP protocol.
7. Draw and explain TCP segment header.
8. Explain TCP connection management.
9. Explain UDP operation.
10. Explain difference between TCP and UDP.
11. Write short note on:
    - TCP
    - UDP
    - DNS

Chapter 12

UPPER LAYERS

INTRODUCTION

Having finished all the preliminaries, we now come to the upper layers. In these we have to study first is the session layer, then presentation layer and lastly the application layer.

However, even in the application layer there is a need for support protocols to allow the real applications to function. Accordingly, we will look, the first area is security, which is not a single protocol, but a large number of concepts and protocols that can be used to ensure privacy where needed. The second is DNS, which handles naming within the Internet. The third support protocol is for network management after that, we will examine the real applications such as electronic mail, USENET (net news), WWW, HTTP, and HTML.

12.1 NETWORK SECURITY

Network security is looming on the horizon as a potentially massive problem. Security is a broad topic and covers a multitude of sins.

Most security problems are intentionally caused by malicious people trying to gain some benefit or harm someone. A few of the most common perpetrators are given in Table 12.1. It should be clear from this list that making a network secure involves a lot more than just keeping it free of programming errors.

Table 12.1 Some people who cause security problems and why

Adversary	*Goal*
Student	To have fun snooping on people's email
Hacker	To test out some one's security system; steal data

Sales rep.	To claim to represent all of Europe, not just-Andorra
Businessman	To discover a competitor's strategic marketing plan
Ex-employee	To get revenge for being fired
Accountant	To embezzle money from a company
Stockbroker	To deny a promise made to a customer by email
Con man	To steal credit card numbers for sale
Spy	To learn an enemy's military strength
Terrorist	To steal germ warfare secrets

Network security problems can be divided roughly into four intertwined areas:

Secrecy

Secrecy has to do with keeping information out of the hands of unauthorized users. This is what usually comes to mind when people think about network security.

Authentication

Authentication deals with determining whom you are talking to before revealing sensitive information or entering into a business deal.

Nonrepudiation and Integrity Control

Nonrepudiation deals with signatures: How do you prove that your customer really placed an electronic order for ten million left-handed doohickeys at 89 cents each when he later claims the price was 69 cents? Finally, how can you be sure that a message you received was really the one sent and not something that a malicious adversary modified in transit or concocted?

Physical Security is the most fundamental of all security levels because it deals with securing your technology equipment from damage or theft, protecting it against accidental power surges, and the like. The responsibility for physical security is in the hands of many.

This level of security is most often left out or a minimal part of technology plans, but is extremely important. Take for instance the case of a school district that is planning to put in an average of 6 computers in each classroom over the duration of 5 years. In year one, the first couple of network connections and computers and arrive in the classrooms and users begin storing their files on local drives and such. As the 3rd or 4th computers arrive latter in the plan, the school starts noticing frequent power outages and data is lost because of the lack of proper uninterrupted power supplies (**UPS**) on the machines and network equipment. It is important to plan for the drain on current energy resources that these new technologies will inevitably render. This is the case in many older school buildings throughut the country, as they find the costs of adding computer are great in terms of the behind-the-scenes resources that need to be expanded to accommodate these new technologies.

Physical security is a vital part of any security plan and is fundamental to all security efforts.

12.2 CRYPTOGRAPHY

Cryptography has a long and colourful history. Historically, four groups of people have used and contributed to the art of cryptography: the military, the diplomatic corps, diarists, and lovers.

Figure 12.1 shows a cryptosystem.

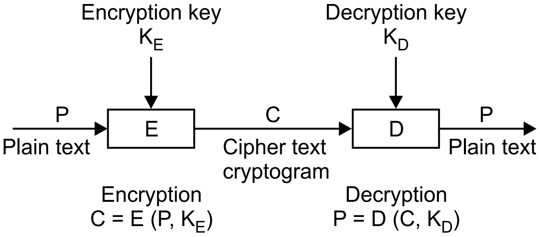


Fig. 12.1 Cryptosystem

A message (P), the "plaintext", is encrypted by a function (E), which is parameterised by a "Key". The resulting encrypted message is called a "Cryptogram" or "Cipher text". The cipher text is sent over the communication link to the receiver. The receiver decrypts the cryptogram by a function that is parametrised by another key to recover the original plaintext P. If the encryption and decryption key are **identical**. ⇒ a **symmetric** or one-key system, *i.e.*, $K = K_E + K_D$. If the encryption and decryption key are different ⇒ **asymmetric** or two-key system, *i.e.*, $K_E \# K_D$.

The art of developing ciphers is called **"Cryptography".** The art of breaking them is called "Cryptoanalyses". Both together are parts of the science of **"Cryptology".**

Substitution Ciphers

In a substitution cipher another letter or group of letters replaces each letter or group of letters.

Example:

Plain text :	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Cipher text :	Q	W	E	R	T	Y	U	I	O	P	A	S	D	F	G	H	J	K	L	Z	X	C	V	B	N	M

Caesar Cipher

Each letter of the plain text is replaced by a letter, which comes K positions later in the alphabet.

The original Caesar Cipher used K = 3, so that a → d, b → e, ... If the letters a – Z are numbered from 1 to 26 (thereby N = 26) than encryption with key K is : E = (P + K) mod N.

Mono Alphabetic Substitution

In a more general approach all letters are mapped via a table. The key is that represented by the character map.

For example: there are 26! = 4 × 10^{26} possibilities to check in order to find the key by a brute force approach. A computer, which checks one possibility each microsecond would still need 1013 years to complete the search.

Breaking Mono-Alphabetic Ciphers

The frequency of letters, two letter combinations, three letter combinations at words in English (and more commonly used) language(s) are well-known. By calculating the frequencies of all letters and combinations in the cipher text a cryptoanalyst will be able to recover the original text without performing an exhaustive search.

Transposition Ciphers

A transposition cipher records the letters in a plain text without substitution. Therefore, the frequency information does not help. *e.g.*, if a key 0 key of K = 4, a possible scheme would arrange the plain text in rows with 4 columns each. The columns are then read one after another to produce the cipher text.

Ex. Plain text : Communication .is. easy

Cipher text : Cuan, yont.emiiiamcoss

Arranging the cipher the different keys K until it reveals the plain text can break this cipher.

Product Ciphers

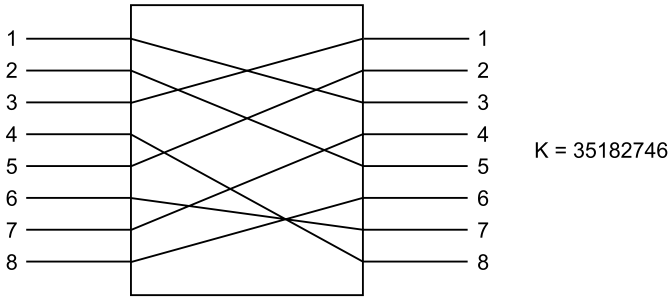


Fig. 12.2 P-Box

The "P" stands for permutation, since it transposes bit positions. A P-Box as shown in Figure 12.2. Wires n-input bits to n-output bits. Text characters are represented in most cases by an 8-bit code (ASCII). The key K describes the new bit positions for each input bit.

If a cryptographer would like to support any substitution for any 8-bit character to any other 8-bit character he would need a table with 256 character entries. The key would have a total length of 256 * 8 = 2048 bits to describe the table. To reduce this a P-Box is encapsulated between a decoder and encoder as shown in Figure 12.3.

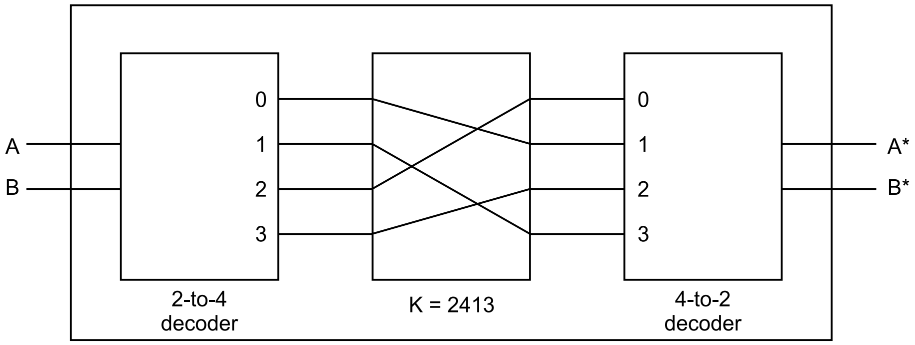


Fig. 12.3 S-Boxes

Product Ciphers cascade units of S- and P-boxes to realise more powerful encryption schemes. An example of a product cipher is DES (data encryption standard), which was released by the U.S. National Bureau of Standard (Now know as the **National Institute of Science and Technology** (NIST) in 1977.

Cryptographic principles are as follows:

- All messages must contain redundancy to prevent active intruders from tricking the receiver into acting on a false message.

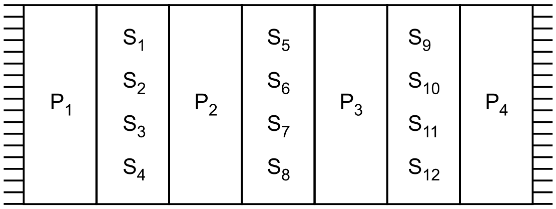


Fig. 12.4 Product cipher

12.3 SECRET KEY ALGORITHMS

Modern cryptography uses the same basic ideas as traditional cryptography, transposition and substitution, but its emphasis is different. Traditionally, cryptographers have used simple algorithm and relied on very long keys for their security. Now-a-days the reverse is true the object is to make the encryption algorithm so complex and involuted that even if the cryptanalyst acuries vast mounds of enciphered text of his own choosing, he will not be able to make any sense of it at all.

Transpositions and substitutions can be implemented with simple circuits. Figure 12.2 show a device, known as P-box, used to effect a transposition on an 8-bit input. If the 8-bits are designed from top to bottom as 01234567, the output of this particular P-box is 36071245. By appropriate internal wiring, a P-box can be made to perform any transposition and do it at practically the speed of light.

Substitutions are performed by S-boxes as shown in Figure 12.3. In this example a 3-bit plain text is entered and a 3-bit cipher text is output. The 3-bit input selects one of the eight lines existing from the first stage and sets it to 1; all the other lines are 0. The second stage is a P-box. The third stage encodes the selected input line in binary again. With the wiring shown. If the eight octal numbers 01234567 were input one after another, the output sequence would be 24506713. In other words, 0 has been replaced by 2, 1 has been replaced by 4, etc. Again, by appropriate wiring of the P-box inside the S-box, any substitution can be accomplished.

The real power of these basic elements only becomes apparent when we cascade a whole series of boxes to form a product cipher, as shown in Figure 12.4. In this example, 12 input lines are transposed by the first stage. Theoretically it would be possible to have the second stage be an S-box that mapped a 12-bit number onto another 12-bit number. However, such a device would need 2^{12} = 4096 crossed wires in its middle stage. Instead, the input is broken up into four groups of 3-bits, each of which is substituted independently of the others. Although this method is less general, it is still powerful.

(DES) Data Encryption Standard

Data encryption standard (DES) is a block-oriented cipher that encrypts blocks of 64-bits. A key of 56-bits controls the encryption. The 56-bit key is transformed into 16 subkeys @ 48-bits to control 16 substitutions in a DES chip, which is illustrated in Figure 12.5.

The encryption process involves 19 stages in a DES chip. The first is a transposition using a fixed transposition rule followed by 16 substitutions and two final transpositions. At each substitutions stage the most and least significant 32-bits of the block are swapped. The former most significant 32-bits are substituted and transposed under subkey control. The result is then forwarded to the next stage.

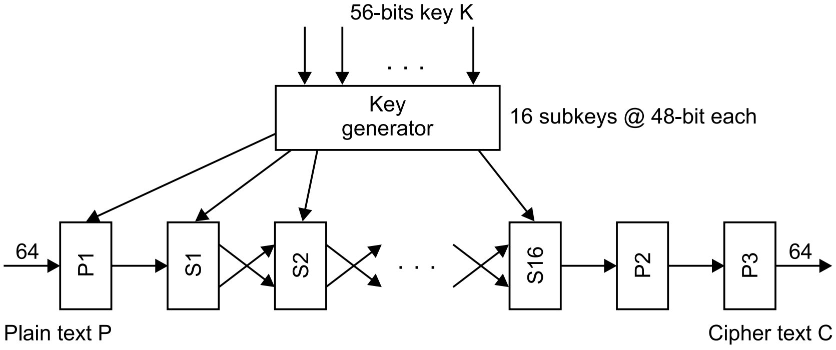


Fig. 12.5 Generating cipher text

Key length

Original proposal by IBM was 128-bit US National Security Agency requested the key sized be reduced to 56-bits. Several proposals used to break messages encrypted with this system.

DES-Operational Modes

- **Electronic Code Book (ECB):** Each block of cipher text is encrypted independently of any other block. Therefore each cipher text block corresponds to one plain text block just like a codebook.
- **Chain Block Cipher (CBC):** ECB does not protect against insertion of repeated blocks because blocks a treated independently. Another weakness is that identical plain text blocks generate identical cipher text blocks.

  To improve DES for communication streams each 64-bit block is EXORed with the previous 64-bit cipher text before entered into the DES chip. In addition to a common secret key the sender and receiver needs to agree on an initial vector to be EXORed with the first block of messages stream.
- **Cipher Feedback Mode (CFM).** CFM is an alternate mode for DES on 8-bit characters. The input character is EXORed with the least significant byte of the DES output and then transmitted over the communication link.

In order to collect enough bits for the 64-bit encryption block the output characters are collected in a character based shift register.

Each output character advances the shift register by 8-bits and triggers a new DES encryption. Thereby the next input character will be EXORed with a new DES output. CFM is suitable for use on serial lines.

12.4 ENCRYPTION WITH PUBLIC KEYS AND PRIVATE KEYS

Key Distribution Problem

DES requires that sender and receiver know a common secret key, the question arises how to distribute keys between communication partner.

Public-key cryptography requires each user to have to keys:

- **A Public Key:** It is used by the entire world for encrypting messages to be sent to that user.
- **A Private Key:** It is used for, the user needs for decrypting messages.

Public key Systems

In 1976 Diffie and Hellman proposed to use a encryption algorithm E and decrytion algorithm D such that:

1. D (E(P)) = P
2. It is exceedingly difficult to deduce D From E.
3. E cannot be broken by a chosen plain text attack. Under these conditions E can be made public while D is kept secret.

Consider a sender A can send a receiver B a secret key by using public key cryptography. First A retrieves B's public encryption key E_B from a public database. With E_B A can now encrypt a message containing the key K for B. Only B possesses its own secrete decryption key D_B to recover the plain text and thereby K.

Since symmetric cryptosystems with a common key K run much faster than asymmetric public key schemes, A and B decide to switch from public key techniques to a common secret key algorithm for the remainder of their session.

Authentication

Authentication is identification plus verification. Identification is the process whereby a person claims a certain identify. While verification is the process whereby the claim is checked.

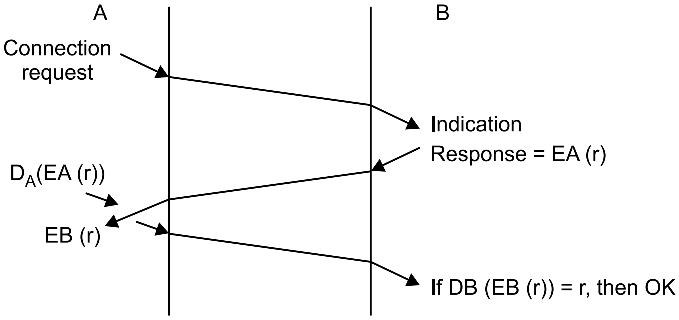


Fig. 12.6 Authentication techniques

An authentication by public keys can be done in the following way. Refer Figure 12.6. A sends a connection request to B. B sends A's "challenge" by encoding a random number *r* with A's public key EA.

Only A is able to decrypt the challenge with his private key DA. He sends B the result openly or encoded with B's public key. B can now verify whether the random number was decoded correctly or not.

RSA Algorithm

Based on the ideas of Diffie and Hellman a public cryptographic scheme was developed by Rivest, Shamir and Adleman in 1978 called RSA. This scheme satisfied the condition that different keys that were not derivable from each other do the encryption E and decryption scheme D.

Raising it to the power E modulo N, where N is a very large number, encrypts a plain text P. The result C and P must be within the range [0, N – 1] due to the modulo function. The decryption is done the same way by raising the cipher text C to the power D modulo N.

The modulo N is based on the product of two large prime numbers which also act as the bases for E and D. To break the code a cryptoanalyst must Factorise N into the two prime numbers again. Depending on the size of N the time needed is so large (N > 200 digits → over million years) that breaking it by brute force becomes uninteresting.

However, things are not necessarily as “difficult” as they seem. In 1977, a 129-digit number named R.S.A. 129-digit was chosen as the basis for a code presumed to be impregnable because breaking the code would require factoring the number—a process that was though to require 40 quadrillion years.

That was a miscalculation, the number was factored by 100 quadrillion calculations, contributed by more than 600 Internet participants. Solution use even larger numbers.

12.5 SESSION LAYER OPERATIONS AND SERVICES

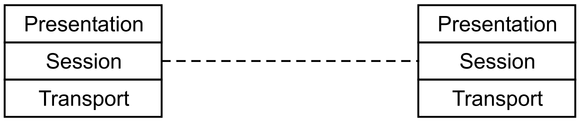


Fig. 12.7 Session layer

The session layer is responsible for requesting a logical connection to be established for the communication process. This shared connection is called session. This layer provides for data synchronisation between user tasks by placing checkpoints so that, in the event of a network failure, only the data sent after the point of failure needs to be resent. This layer also provides services such as name lookup. Security, allowing two programs to find each other and establishing the communication link. This layer controls the dialog between TV processes, determining who can transmit and who can receive at what point of time during the communication.

Example: Consider a firm that has a clerk who handles only incoming mail (“receiving”) and a clerk who handles only outgoing mail (“shipping”). Each clerk, therefore, implements one end of the transport layer. Suppose they report to a chief clerk who not only accepts letters for transmission from the firm’s employees, but also delivers incoming mail to them. The chief clerk, therefore, plays the role of a session layer, managing two simplex connections to present the abstraction of a duplex connection. The chief clerk may also expedite some letters (using courier service instead of regular service). Finally, if the clerk is given a set of letters such that either all or none must be delivered. (Such as invitations to an important meeting), she or he can arrange with her or his peer to discard incomplete sets to provide this abstraction.

The internet does not have a standard session layer protocol TCP already provides duplex and expedited data delivery, and session rollbacks are part of the application, if necessary.

Functions of the session layer:

- Association of sessions with transport connections.
- Flow control for the session.
- Performing data exchange between tasks.
- Opening, terminating and re-establishing session connections.
- Session layer management and communication with adjacent layers.

- Dialogue control (who, when, how long, half- or full-duplex).
- Recovery from communication problems during a session without data loss.

Services provided by the session layer:

- Establishing and terminating the session.
- Performing data transfer.
- Dialogue control.
- Synchronisation of the session connection.
- Notification of unrecoverable errors.

12.6 ASN. 1 ABSTRACT SYNTAX NOTATION 1

A standard object definition language, along with encoding rules, used by SNMP is taken from OSI and is called as ASN. 1 (**Abstract Syntax Notation One**). It is similar to OSI, it is large, complex, and no especially efficient.

For better or worse, SNMP is drenched in ANS. 1, so anyone wishing to truly understand SNMP must become fluent in ASN. 1. The ASN. 1 abstract syntax is essentially a primitive data declaration language. It allows the user to define primitive objects and then combine them into more complex ones. A series of declarations in ASN. 1 is functionally similar to the declarations found in the header files associated with many C programs.

The ASN. 1 basic data types allowed in SNMP are shown in Table 12.2. The use of the code will be described later.

Table 12.2 The ASN-1 Primitive data types permitted in SNMP

Primitive type	*Meaning*	*Code*
INTEGER	Arbitrary length integer	2
BIT STRING	A string of 0 or more bits	3
OCTET STRING	A string of 0 or more unsigned bytes	4
NULL	A place holder	5
OBJECT IDENTIFIER	An officially defined data type	6

ASN-1 Transfer Syntax

An ASN-1 transfer syntax defines how values of ASN. 1 types are unambiguously converted to a sequence of bytes for transmission. The transfer syntax used by ASN. 1 is called **BER** (Basic Encoding Rules). ASN-1 has other transfer syntaxe that SNMP does not use. The rules are recursive, so the encoding of a structured object is just the concatenation of the of the encodings of the component objects. In this way, all object encodings can be reduced to a well-defined sequence of encoded primitive objects. The encoding of these objects, in turn, is defined by the BER.

The guiding principle behind the basic encoding rules is that every value transmitted, both primitive and constructed ones, consists of up to four fields:

1. The identifier (type or tag).
2. The length of the data field, in bytes.
3. The data field.
4. The end-of-contents flag, if the data length is unknown.

The last one is permitted by ASN-1, but specifically forbidden by SNMP, so we will assume the data length is always known.

The first field identifies the item that follows. It, itself, has three subfields, as shown in Figure 12.8. The higher order 2-bits identify the tag type. The next bit tells whether the value is primitive (0) or not (1). The tag bits are 00, 01, 10 and 11, for UNIVERSAL, APPLICATION, Context-Specific and PRIVATE, respectively. The remaining 5-bits can be used to encode the value of the tag if it is in the range 0 through 30. If the tag is 31 or more, the low-order 5-bits contain 11111, with the true value in the next byte or bytes.

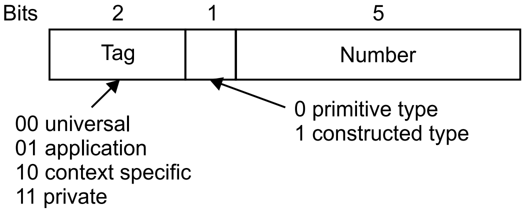
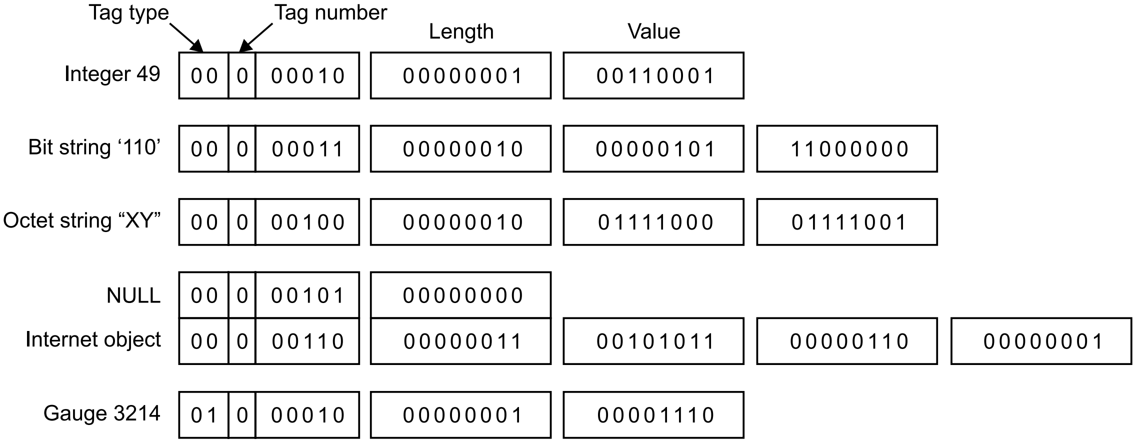


Fig. 12.8 The 1st byte of each data item sent in the ASN. 1

The encoding of the data field depends on the type of data present. Integers are encoded in two's complement. Bit strings are encoded as themselves. The only problem is how to indicate the length. The length field tells how many **bytes**—the value has, not how many **bits**. The solution chosen is to transmit 1-byte before the actual bit string telling how many bits (0 thro' 7) of the final byte are unused. Thus the encoding of the *g*-bit string '010011111' would be 07, 4F, 80 (hexadecimal).



				Length	Value			
Integer 49	00	0	00010	00000001	00110001			
Bit string '110'	00	0	00011	00000010	00000101	11000000		
Octet string "XY"	00	0	00100	00000010	01111000	01111001		
NULL	00	0	00101	00000000				
Internet object	00	0	00110	00000011	00101011	00000110	00000001	
Gauge 3214	01	0	00010	00000001	00001110			

Fig. 12.9 ASN-1 encoding of some example values

Octet strings are easy. The null value is indicated by setting the length field to 0. No numerical value is actually transmitted.

An object identifier is encoded as the sequence of integers it represents.

An example showing some encoding values is given by 12.9.

12.7 SNMP (SIMPLE NETWORK MANAGEMENT PROTOCOL)

When the ARPANET turned into the World Wide Internet, with multiple backbones and multiple operators, this solution ceased to be adequate, So better tools for network management were needed. Two early attempts were defined in RFC 1028 and RFC 1067, but these were short lived. In May 1990, RFC 1057 was published, defining version 1 of **SNMP** (**Simple Network Management Protocol**). Along with a companion document (RFC 1155) on management information, SNMP provided a systematic way of monitoring and managing a computer network. This framework and protocol were widely implemented in commercial products and became the de facto standards for network management.

Although SNMP was designed with the idea of its being simple. The SNMP model of a managed network consists of four components:

1. Managed nodes.
2. Management stations.
3. Management information.
4. A management protocol.

The SNMP model is shown in Figure 12.10

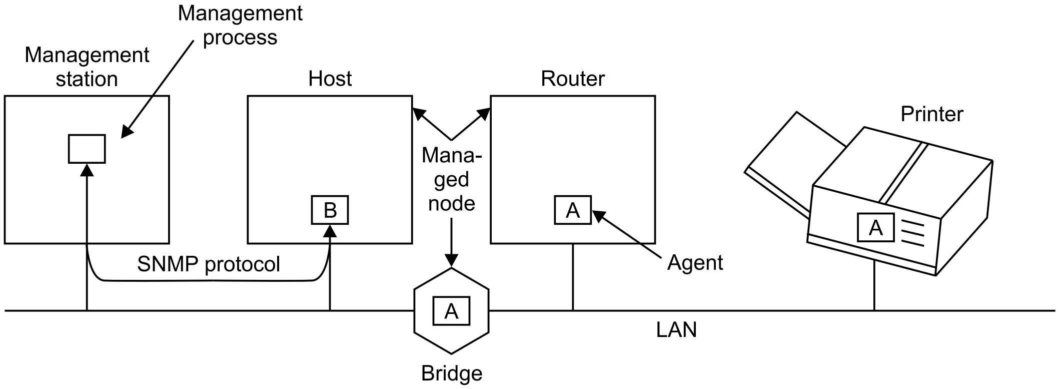


Fig. 12.10 Components of the SNMP management model

The **managed nodes** can be hosts, routers, bridges, printers, or any other devices capable of communicating status information to the outside world. To be managed directly by SNMP, a node must be capable of running an SNMP management process, called an **SNMP agent**. All computers meet this requirement, as do increasingly many bridges, routers, and peripheral devices designed for network use. Each agent maintains a local database of variables that describe its state, history and affect its operation.

Network management is done from **management stations**, which are, in fact, general-purpose computers running special management software. The management stations contain one or more processes that communicate with the agents over the network, issuing commands and getting responses. In this design, all the intelligence is in the management stations in order to keep the agents as simple as possible and minimize their impact on the devices they are running on. Many management stations have a graphical user interface to allow the network manager to inspect the status of the network and take action when required.

In order to allow a management station to talk to all these diverse components, the nature of the information maintained by all the devices must be rigidly specified. The SNMP describes the exact information each kind of agent has to maintain and the format it has to supply it in.

Very briefly, each device maintains one or more variables that describe its state. In the SNMP literature, these variables are called **objects**, but the term is misleading because they are not objects in the sense of an object-oriented system because they just have state and no method. Nevertheless, the term is so ingrained (*e.g.*, used in various reserved works in the specification language used) that we will use it here. The collection of all possible objects in a data structure called the **MIB** (Management Information Base).

The management station interacts with the agents using the SNMP protocol. This protocol allows the management station to query the state of an agent's local objects, and change them if necessary. Most of SNMP consists of this query-response type communication.

Older devices or devices not originally intended for use on a network may not have this capability. To handle them, SNMP defines what is called **a proxy agent**—namely an agent that watches over. One or more nonSNMP devices and communicate with the management station on their behalf, possibly communicating with the devices themselves using some nonstandard protocol.

Finally, security and authentication play a major role in SNMP.

12.8 DNS (DOMAIN NAME SYSTEM)

When thousands of workstations were connected to the net, every one realized that this approach could not continue to work forever. For one thing, the size of the file would become too large. However, even more important, host name conflicts would occur constantly unless names were centrally managed. Something unthinkable in a huge international network. To solve these problems, DNS (the Domain Name System) was invented.

The essence of DNS is the invention of a hierarchical, domain-based naming scheme and a distributed database system for implementing this naming scheme. It is primarily used for mapping host names and email destinations to IP addresses but can also be used for other purposes. DNS is defined in RFCs 1034 and 1035.

Briefly, the DNS is used for mapping a name onto an IP address, an application program calls a library procedure called the **resolver**, passing it the name as a parameter. The resolver sends a **UDP** packet to a local DNS server, which then looks up the name and returns the IP address to the resolver, which then returns it to the caller. Armed with the IP address, the program can then establish a TCP connection with the destination, or send it UDP packet.

The DNS Name Space

Managing a large and constantly changing set of names is a nontrivial problem. Conceptually, the Internet is divided into several hundred top-level **Domains**, where each domain covers many hosts. Each domain is partitioned into sub-domains, and these are further partitioned, and so on. All these domains can be represented by a tree, as shown in Figure 12.11. The leaves of the tree represent domain that have no subdomains. A leaf domain may contain a single host, or it may represent a company and contains thousands of hosts.

The top-level domains come in two flavours **generic** and **countries**. The generic domains are com (commercial), edu (educational institutions), gov (the US federal government), int (certain International organizations). The country domains include one entry for every country, as defined in ISO 3166.

Each domain is named by the path upward from it to the root (unnamed). The component are separated by periods (pronounced "dot"). This hierarchical naming means that **eng.sun.com** does not conflict with a potential use of eng in **eng.yale.edu.**, which might be used by the Yale English department.

Domain names can be either **absolute** or **relative**. An **absolute** domain name ends with a period (*e.g.*, eng.sun.com), whereas a relative one does not. Relative names have to be interpreted in some context to uniquely determine their true meaning. In both cases, a named domain refers to a specific node in the tree and all the nodes under it.

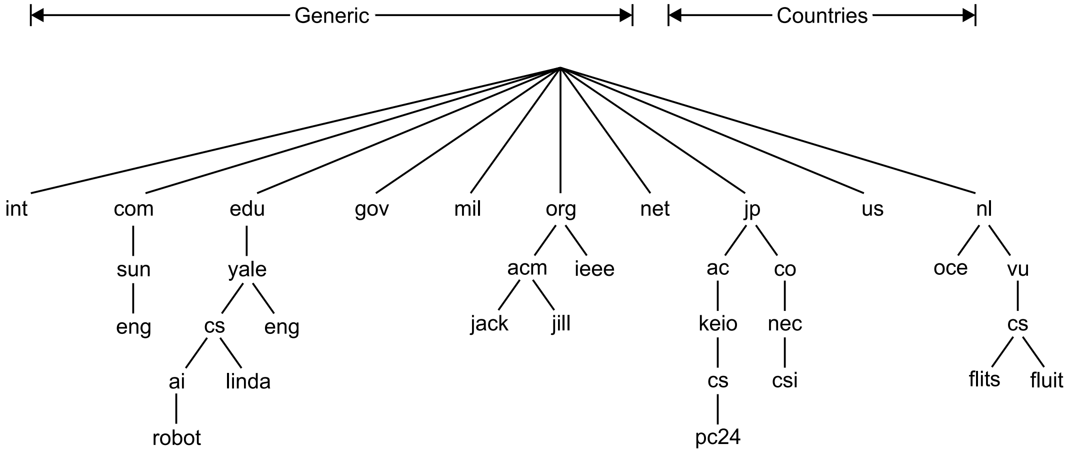


Fig. 12.11 A portion of the internet domain name space

Domain names are case insensitive, so edu and EDU mean the same thing. Component name can be to 63 characters long, and full path names must not exceed 255 characters.

Domains can be inserted into the tree in two different ways. For example, cs.yale.edu could equally well be listed under the **US** country domain as cs.yale.ct.us.

Each domain controls how it allocates the domains under it. For example, Japan has domains ac.jp and co.jp that mirror edu and com. The **Netherland** does not make this distinction and puts all organizations directly under nl.

To create a new domain, permission is required of the domain in which it will be included for example, if a **VLSI** group is started at Yale and wants to be known as VLSI. cs.yale.edu, it needs permission from whomever manages cs.yale.edu.

Resource records

Every domain, whether it is a single host or a top-level domain, can have a set of **resource** records associated with it. For a single host, the most common resource record in just it IP address, but many other kinds of resource records also exist. When a resolver gives a domain name to DNS, what it gets back are the resource records associated with that name. Thus the real function of DNS is map domain names onto resources records.

A resource record is a five-tuple. Although they are encoded in binary for efficiency, in most expositions resource records are presented as ASCII test, one line per resource record. The format we will use is as follows:

Domain-name Time-to-live Type Class Value.

Domain-Name

The domain-name tells the domain applied to this record. Normally, many records exist for each domain and each copy of the database holds information about multiple domains. This field is thus the primary search key used to satisfy queries.

Time-to-live

This field gives an indication of how stable is the record. Information that is highly stable is assigned a large value, such as 86400 (the number of seconds in 1 day). Information that is highly volatile is assigned a small value, such as 60 (1 minute).

Type

The type field tells what kind of record this is. The most important types are given in Table 12.3.

Table 12.3 The principal DNS resource record type

Type	*Meaning*	*Value*
SOA	Start of Authority	Parameters for this zone
A	IP address of a host	32-bit integer
MX	Mail exchange	Priority, domain willing to accept email
NS	Name serve	Name of server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
HINFO	Host description	CPU and OS in ASCII
TXT	Text	Uninterpreted ASCII Text

Class

The fourth field of every resource record in the class. For Internet information, it is always IN. For non-Internet information, other codes can be used.

Value

Finally, we come to the value field. This field can be a number, a domain name, or an ASCII string. The semantics depend on the record type. A short description of the value field for each of the principal records type is given in Table 12.3.

Records's Service

1. **SOA:** An SOA record provides the name of the primary source of information about the name server's zone, the email address of its administrator, a unique serial number, various flags and timeouts.
2. **A:** A denotes an (Address) record. It holds a 32-bit IP address for some host.
3. **MX:** It specifies the name of the domain prepared to accept email for the specified domain.
4. **NS:** The NS records specify name servers.

12.9 APPLICATION LAYER SERVICES

The application layer may provide following services:

1. Identification of the communication partners and establishing availability.
2. Authorisation and validity checks.
3. Allocation of costs.
4. Agreeing available resources.
5. Accepting services.
6. Synchronising applications.
7. Error correction.
8. Selecting the dialogue.
9. Checking data integrity.
10. Complying with the data syntax.
11. File request and file transfers.
12. Down line loading.
13. Graphics procedures.
14. Database queries, insertions and deletions.
15. Virtual terminal service.
16. Job manipulation and remote job entry.
17. Electronic mail.

12.10 STUDY OF INTERNET TOOLS

Now we will discussing some of the Internet tools such as E-mAIL, USENET news, WWW, HTTP, HTML.

12.10.1 E-mail

The Concept of E-mail

The most common use of the Internet is electronic mail (popularly known as e-mail). Using e-mail a user can send text, pictures, sound files, program files, or animated movies to any other person on the network any where in the world.

There are two main advantages of using e-mail:

1. The speed at which delivery takes place (almost instantaneously).
2. It costs almost the same regardless of the distance.

Working of E-mail

Every Internet mail user has a unique Internet e-mail address. This e-mail address is in the format-username @ domain name.

Figure 12.12 shows how mail messages move across the Internet.

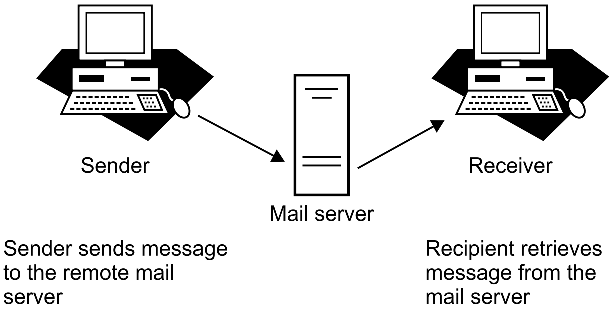

Sender
Mail server
Receiver
Sender sends message to the remote mail server
Recipient retrieves message from the mail server

Fig. 12.12 An e-mail system

Following steps are involved in sending an e-mail message:

1. The sender composes the mail message using his mail client software.

   A mail client allows a user to compose, edit and send the mail message. There are a number of mail client software available. Netscape Mail, Outlook express, Eudora, Pine, etc. are the examples of mail clients.

2. After composing the mail message the user sends it to the recipient's e-mail address.

   The message propagates across the Internet before it reaches the mail server of the recipient. The domain name in the recipient's e-mail address identifies his mail server and the username identifies the recipient on that server.

3. The recipient connects to his e-mail account on his mail server to read the message sent to him.

   These recipient also uses a mail client to receive, save, and print mail message.

Internet Mail Protocols

Internet e-mail is based on standards such as Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP), Internet Message Access Protocol (IMAP), and Multipurpose Internet Mail Extensions (MIME). These standards are known as mail protocol and these are described below.

The simple mail transfer protocol (SMTP)

The SMTP is a standard protocol used to transfer mail messages between computers.

The SMTP specifies the format in which a mail message has to be composed.

SMTP uses the ASCII character set for composing a message. An Internet mail message has two parts—a header and a body. The header of the message includes the address of the recipient, address of the sender, subject, and other information about the message such as date and time it was sent, type of the mailing client the sender is using, etc. The body of the message contains the actual message.

The post office protocol (POP)

The Post Office Protocol defines how mail clients can retrieve messages from a mail server. The POP was developed for single user computers. There are three versions of this protocol: POP, POP2 and POP3.

The internet message access protocol (IMAP)

Developed at Stanford University in 1986, the Internet Message Access Protocol is for retrieving e-mail messages. The latest version, IMAP4, is similar to POP3 but supports some additional features. For example, with IMAP4, you can search through your e-mail messages for keywords while the messages are still on mail server. You can then choose which messages to download to your machine. Like POP, IMAP uses SMTP for communication between the e-mail client and server.

The multipurpose internet mail extensions (MIME)

The SMTP can be used to send only messages that are composed using ASCII character set. This restricts the utility at electronic mail. There is another protocol MIME that can be used to exchange e-mail messages containing non-textual data such as graphics, sound, and other multimedia files.

Multipurpose Internet Mail Extensions (MIME) offers a way to extend Internet standard mail so that users can interchange text in languages with different character sets, and multimedia electronic mail among different computer systems that implemented Internet standard mail.

Whenever you want to send a non-text file such as a spreadsheet, program file, graphics file, or a sound file, you can encode this file using MIME. The MIME encodes these files in a textual form, which can be sent using the SMTP. The recipient can then decode this MIME-encoded data to the original non-text file.

Most e-mail clients such as Netscape automatically encode and decode the e-mail message containing non-text data. However, there are some e-mail clients which require you to encode data using an encoding utility such as Infoxfer.

12.10.2 USENET News

Usenet is a worldwide hierarchical message system. It was started back in 1980 by four students on the university campus and has grown to become the largest decentralized information repository in existence. Now following are some usenet features:

1. Usenet is not controlled or policed by anyone as a whole. No one has authority over usenet, its users define what does and does not occur. No government owns or controls usenet, other than standard national security restrictions. The owners of a local server have control over their server, it is up to them to determine was is or isn't transmitted over their site. Individual users and moderators often try to regulate or control the content of specific newsgroups in order to maintain quality for all participants in a group with relative success.
2. Usenet is not an organization, company, or business other than the standards required to maintain universal compatibility between news servers, there is no wide spread control over how usenet functions are used.
3. Usenet is not fair, since there are no governing or controlling bodies restricting activities, anyone can post anything anywhere they want.
4. Usenet access is not part of the freedom of speech. The owners of the computers which use and transmit usenet can and do restrict what is transmitted to or from their servers. If you want to have complete control over what you can say and read by your own server.
5. Usenet is not a publicly funded entity. It is funded by whatever means necessary to keep existing servers running to add servers as needed.
6. Usenet is not commercial, even though most hosts are now owned by business, blatant wide spread commercial advertising is often shunned. Usenet encompasses government agencies, educational facilities, businesses of all sizes, and individual home computers.
7. Usenet is not the Internet. The Internet is a loose conglomeration of many information exchange, repository and retrieval systems, only one of which is usenet. Usenet can and does exit on servers which are not on the Internet. However, the Internet is an indispensable medium which usenet travels over, if the Internet did not exist, neither would usenet.
8. Usenet is not only in the United States. Like the Internet, usenet is world wide and is limited only by transmission lines and servers. Most of the usenet's newsgroups can be accessed from all of the seven continents, including Antarctica.

9. Usenet is the collection of people who post articles to newsgroups. Usenet is not just the bits, the millions and millions of bits transmitted daily, it is the collection of people who care enough to send the very best. Anyone from anywhere with anything to say about any subject with a connection to any communication network with usenet access can post and retrieve articles.

   Usenet is the ultimate equal opportunity employer.

10. Usenet is not just one messaging system, but is many systems tied together which interact with and without human involvement. Usenet can be accessed by e-mail, FTP, gopher, Web and even Telnet, and that doesn't even cover all the methods possible.

Following are some of the **newsgroups:**

- news.announce.newusers
- news.announce.newgroups
- news.newusers.questions
- news.answers
- news.groups

These newsgroups contain important usenet-related FAQs and other information postings usenet has over 15,000 newsgroups and more are constantly being added. Usenet is loosely organized into nine major categories.

• alt	limited distribution groups, often with a wider range of content
• biz	business
• comp	computers, software
• misc	anything that doesn't fit into another category
• rec	recreation and entertainment
• news	news and topical subjects
• sci	science
• soc	social issues and socializing
• talk	debate and discussion and also other categories exists.

Almost every newsgroup has a FAQ (Frequently Asked Questions), which is a document containing the focus, purpose, posting limitations and rules for the specific newsgroups.

A few alternate locations for usenet FAQ are:

- Usenet Graphics FAQs Web page. This site contains many graphical newsgroups and other related FAQs.

  http://www.cis-chio-state.edu/hyper-text/faq/usenet/graphics/top.hton
- usenet.newsgroups:Resources

  http://www.ucs.india.edu/netRSC/usenet.html
- MITFAQ FTP archieve

  ftp://rtfm.mit.edu/pub/usenet/news/news.answers

12.10.3 WWW (World Wide Web)

The acronym WWW stands for World Wide Web. The World Wide Web is officially described as a “wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.”

The term hypermedia has been derived from hypertext. Hypertext is text that is non-sequential or non-linear in nature. Hypertext describes the ability to link related documents together using words and phrases. Hypermedia is a natural extension of hypertext in that the contents of each document include much more than text. They include multimedia (images, sounds and video).

WWW provides a consistent means to access a variety of information in a simplified manner to the users on computer networks.

Tim Berners-lee, a research scientist at the European particles Physics Laboratory (CERN) in Geneva, Switzerland, developed the concept of WWW in 1989.

The earlier WWW systems were designed for two main purpose. Advancement of science and education. But, WWW has made a significant impact not only on these two areas, but many other such as commerce, politics and also literature.

There is no standard way of viewing it or navigating around the WWW. However, many software interfaces to the web have similar functions and generally work the same way no matter what computer or type of display is used. There are a number of web browsers such as Microsoft Internet Explorer, Netscape Navigator, Mosaic, etc. that can be used to access the information on the Internet. All of these are graphical browsers. In addition, most modern browsers can present multimedia information, including sound and video.

Although there are many different ways to represent a document on the screen, it is often called a **page.** The web page designers create a special document that is intended to be viewed first-one that contains introductory information and/or a master menu of documents within that collection. This type of document is called a **home page** and is generally associated with a particular site, person, or named collection. The location at which the web pages are stored is called **web site.** You can access a web site by typing the name of the site.

The WWW system is based on the client server architecture. A web client is used to send request for information to any web server that stores the requested information. A web server is a program that, upon receipt of a request, sends the document requested back to the requesting client.

Typically the client program (*i.e.*, browser) runs on a separate machine from that of the server. The server takes care of all the issues related to document storage where as the task of presenting the information to the user is left to the client program.

Web clients and servers can communicate with each other by the protocol is called the **Hypertext Transfer Protocol**. (Known as HTTP). All web clients and servers must use HTTP in order to send and receive hypermedia documents. For this reason web servers are often called HTTP servers.

Web Page and Hypertext Markup Language (HTML)

A web page is a hypermedia document. The HTML is the standard language used for creating the web pages. The HTML provides a number of commands that can be used to place and format text, pictures, and sound on web pages.

Web documents are typically written in HTML, which is very easy to use, using any text editor such as Window Notpad and are usually named with the suffix.html or .htm. Thus HTML documents are standard ASCII files with formatting codes that contains information about layout and links. After entering the HTML code in a file you can view it using any browser. The browser automatically interprets the HTML code and formats the document accordingly.

Universal Resource Locators

The World Wide Web uses Universal Resource Locators (URLs) to represent hypermedia links and links to network services within HTML documents. It is possible, to represent almost any file or service on the Internet with a URL looks like.

http://www.microsoft.com

The first part of the URL (before the two slashes) specifies the method of access. The second is typically the address of the computer on which the information or service is to be located. Further parts may specify the names of files, the port to connect to, or the text to search for in a database. A URL is always a single line with no embedded spaces.

Searching For Information On The Internet

There are some special web-sites on the Internet, which allow you to search for the desired information on the Internet. Most of these sites also group the information available on the Internet under various categories such as business, sports, entertainment, etc. These categories are also called subject guides. Subject guides are very useful for browsing general topics.

These special web sites are called search engines *e.g.*, Yahoo, Webcrawler, Altavista, Excite, Infoseek and Lycos. Search engine maintains a database of keywords. When you request the search engine for information, it attempts to locate the specified information in its database of keywords. If the search engine finds a match, it displays these references.

12.10.4 HTTP (Hyper Text Transfer Protocol)

The standard web transfer protocol is HTTP (Hyper Text Transfer Protocol). Each interaction consists of one ASCII request, followed by one RFC 822 MIME like response. Although the use of TCP for the transport connection is very common, it is not formally required by standard. If ATM networks become reliable enough, the HTTP requests and replies could be carried in AAL 5 message as well.

HTTP is constantly evolving several versions are in use and others are under development. The material presented below is relatively basic and is unlikely to change in concept, but some details may be a little different in future versions.

The HTTP protocol consists of two fairly distinct items; the set of requests from browsers to several servers and the set of responses going back the other way. We will now treat each of these in turn.

As the newer versions of HTTP-support two kinds of requests.

Simple Requests

A simple request is just a single GET line naming the page desired, without the protocol version. The response is just the raw page, with no headers, no MIME and on encoding. To see how this works, try making a telnet connection to port 80 of WWW.WS.Org and the type ***GET/hypertext/WWW/the project.html***: but without the HTTP/1.0 this time the page will returned with no indication of its content type. This mechanism is needed for backward compatibility. Its use will decline as browsers and servers based on full requests become standard.

Full Requests

Full requests are indicated by the presence of the protocol version on the GET request line. Requests may consists of multiple lines, followed by blank line to indicate the end of the request. The first line of a full request contains the command, the page desired, and the protocol version. Subsequent lines contain RFC 822 headers.

Although HTTP was designed for use in the web, it has been intentionally made more general than necessary with an eye to future object-oriented applications. For this reason, the first word on the full request line is simply the name of the method (command) to be executed on the web page. The built-in methods are listed in Table 12.4. When accessing general objects, additional object-specific methods may also be available. The names are case sensitive, so GET is a legal method but get is not

Table 12.4 The built-in HTTP request methods

Method	*Description*
GET	Request to read a web page.
HEAD	Request to read a web page's header.
PUT	Request to store a web page.
POST	Append to a named resource.
DELETE	Remove the Web page.
LINK	Connects two existing resources.
UNLINK	Break an existing connection between two resources.

GET

GET method requests the server to send the page (by which we mean object, in the most general case), suitably encoded in MIME. However, if the GET request is followed by an **IF-Modified-Since** header, the server only sends the data if it has been modified since the date supplied. Using this mechanism, a browser that is asked to display a cached page can conditionally ask for it from the server, giving the modification time associated with the page. If the each is still valid, the server just sends back a status line announcing that fact, thus eliminating the overhead of transfering the page again.

PUT

The PUT method is the reverse of GET: instead of reading the page, it writes the page. This method makes it possible to build a collection of web pages on a remote server. The body of the request contains the page. It may be encoded using MIME, in which case the lines following the PUT might include content type and authentication headers, to prove that the caller indeed has permission to perform the requested operation.

HEAD

The HEAD method just asks for the message header, without the actual page. This method can be used to get a page's time of last modification, to collect information for indexing purposes, or just to test a URL, for validity. Conditional HEAD requests do not exist.

POST

Some what similar to PUT is the POST method. It too bears a URL, but instead of replacing the existing data, the new data is "appended" to it in some generalized sense. Posting a message to a news group or adding a file to a bulletin board system are examples of appending in this context. It is clearly the intention here to have the web take over the functionality of the USENET news system.

DELETE

DELETE does what you might expect, it removes the page. As with PUT, authentication and permission play a major role here. There is no guarantee that DELETE succeeds, since even if the remote HTTP server is willing to delete the page, the underlying file may have a mode that forbids the HTTP server from modifying or removing it.

LINK, UNLINK

The LINK and UNLINK methods allow connections to established between existing pages or other resources.

Every request gets a response consisting of a status line, and possibly additional information. The status line can bear the code 200 (OK), or any one of a variety of error codes, for example, 304 (not modified), 400 (bad request), or 403 (forbidden).

The HTTP standards described message headers and bodies in considerable detail. Suffice it to say that these are very close to REC 822 MIME messages so we will not look at them here.

12.10.5 HTML (Hyper Text Markup Language)

HTML is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language), but specialized to hypertext and adapted to the web. HTML is a language for describing how documents are to be formatted. HTML thus contains explicit commands for formatting. The **advantage** of a markup language over one with no explicit markup is that writing a browser for it is straight forward the browser simply has to understand the markup commands.

Like HTTP, HTML is in a constant state of flux. When mosaic was the only browser, the language in interpreted, HTML 1.0, was the de facto standard. When new browsers came along, there was a need for a format Internet standard, so the HTML 2.0 standard was produced. HTML 3.0 was initially created as a research effort to add many new features to HTML 2.0, including tables, toolbars, mathematical formulas, advanced style sheets and more.

Below we will give an introduction to HTML, just to give an idea what it is like. While it is certainly possible to write HTML documents with any standard editor, and many people do, it is possible to use special HTML editors that do most of the work.

Basic HTML document can be divided into three by the ⟨BODY⟩ and ⟨/BODY⟩ tags. The commands inside the tags are called directives.

Example:

⟨HTML⟩

⟨HEAD⟩ ⟨TITLE⟩ BSNL ⟨/TITLE⟩ ⟨/HEAD⟩

⟨BODY⟩ ⟨H1⟩ WELCOME TO BSNL HOME PAGE ⟨/H1⟩

⟨/BODY⟩

⟨/HTML⟩

Common HTML tags are given below. Some of them having additional parameters.

Table 12.5 HTML tags

Tag	*Description*
⟨HTML⟩ ....⟨/HTML⟩	Declares the web page to be written in HTML
⟨HEAD⟩ ....⟨/HEAD⟩	Delimits the page's head
⟨TITLE⟩ ....⟨/TITLE⟩	Defines the title
⟨BODY⟩ ....⟨/BODY⟩	Delimits the page's body
⟨Hn⟩ ....⟨/Hn⟩	Delimits a level n heading
⟨B⟩ ....⟨/B⟩	Set....in bold face
⟨I⟩ ....⟨/I⟩	Set....in italics
⟨UL⟩ ....⟨/UL⟩	Brackets an unordered list
⟨OL⟩ ....⟨/OL⟩	Brackets a numbered list
⟨MENU⟩ ....⟨/MENU⟩	Brackets a menu of ⟨LI⟩ idems
⟨LI⟩	Start of a list item

⟨BR⟩	Force a break here
⟨P⟩	Start of paragrah
⟨HR⟩	Horizontal rule
⟨PRE⟩....⟨/PRE⟩	Preformatted text: do not reformat
⟨IMG SRC = “....”⟩	Load an image here
⟨A HREE = “....”⟩....⟨/A⟩	Defines a hyperlink
⟨Marquee⟩	Scrolling the text across your page.

Main sections, as ⟨HTML⟩, ⟨HEAD⟩ and ⟨BODY⟩ which is shown in Figure 12.13.

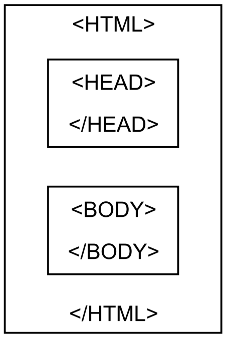


Fig. 12.13 Basic HTML document

The given figure shows the division of HTML document. Each section has beginning and end identifiers. The largest section is represented by ⟨HTML⟩ and ⟨/HTML⟩ where ⟨HTML⟩ is the beginning identifier and ⟨/HTML⟩ is the end identifier.

The identifier ⟨HEAD⟩ and ⟨/HEAD⟩ corresponds to head of the document. The “head” section contains “title” of that document.

The third section is represented by ⟨BODY⟩ and ⟨/BODY⟩. This section forms the body of HTML document and contains the texts, images and various other data. It is the actual information which a “creater” wants to convey to the viewers. You can insert tags in uppercase and lowercase.

A proper web page consists of a head and body enclosed by ⟨HTML⟩ and ⟨/HTML⟩ tags. (formatting commands), although the most browsers donot complaint if these tags are missing. As can be seen in Table 12.5. The head is bracketed by the ⟨HEAD⟩ and ⟨/HEAD⟩ tags and the body is bracketed.

QUESTIONNARIES

1. Explain encryption and decryption with public key and private key.
2. What is public key and private key.
3. Explain various types of cryptography.

4. What are IEEE data encryption standards.

5. Explain RSA algorithm.

6. Explain DES algorithm.

7. Explain different operations performed by session layer.

8. Explain ASN. 1.

9. Write short note on:

- Cryptograph
- Network security
- Session layer operations and Services.

APPENDIX A

BIOS-COM & _BIOS_SERIALCOM

<BIOS.H>

RS-232 communications (serial I/O)

Declaration

- Int bioscom(int cmd,char abyte, int port);
- Unsigned_bios_serialcom(int cmd, int port,char byte);

Remarks

Both bioscom & _bios_serialcom use BIOS interrupt 0x14 to perform various RS-232 communications over the I/O port given in port.

Arg.	**What It Is/Does**
Abyte	OR combinations of bits that specifies COM port settings (ignored if cmd=2 or 3)
Cmd	specifies the I/O operations to perform
Port	Identifies the I/O port; 0 = COM1, 1 = COM2, etc

Return Value

For all values of cmd, both functions return a 16-bit integer.

The upper 8 bits of the return value are status bits.

- If one or more status bits is set to 1, an error has occurred.
- If no status bits are set to 1, the byte was received without error.

The lower 8 bits of the return value depend on the value of cmd specified:\

Value of cmd	**Lower 8 bits of return value**
0(-COM_INIT)or	The lower bits are defined as shown in the preceding diagram.

3(_COM_STATUS)	
1(_COM_SEN)	—
2(_COM_RECEIVE)	The byte read is in the lower bits of the return value—if there is no error (no upper bits are set to 1).

BIOSCOM EXAMPLES

Example 1

/* CHARACTER TRANSMISSION BY SIMPLEX METHOD (SENDER) */

```
#include<iostream.h>
#include<dos.h>
#include<bios.h>
#include<conio.h>
#define com1 0
#define SETTINGS (0x80 | 0x02 | 0x00 | 0x00)

void main()
{
    clrscr();
    char a;

    /* SETTING THE PORT */

    bioscom(0,SETTINGS,com1);
    cout << “Enter the character to transfer:”;
    a = getch();

    /* SENDING DATA FROM PORT*/

    bioscom(1,a,com1);
    getch();
}
/* CHARACTER TRANSMISSION BY SIMPLEX METHOD (RECEIVER) */

#include<iostream.h>
```

```
#include<dos.h>
#include<bios.h>
#include<conio.h>
#define com1 0
#define SETTINGS (0x80 | 0x02 | 0x00 | 0x00)

void main()
{
    clrscr();
    char a;

    /* SETTING THE PORT */

    bioscom(0,SETTINGS,com1);
    while(1)
    {
        char b = bioscom(2,0,com1);
        b = b&0x00ff;
        a=b;
        sleep(3);
        cout << a;
    }
}
```

Example 2

/* CHARACTER TRANSMISSION BY FULL DUPLEX METHOD */

```
#include<stdio.h>
#include<conio.h>
#include<bios.h>
#include<time.h>
#include<iostream.h>
#include<fstream.h>
#define C0M1 0
#define COM2 1
#define SETTINGS ( 0x80 | 0x02 | 0x00 | 0x00)
```

```
void main()
{
char a,s;
int i;
clock_t start,end;
clrscr();
bioscom(0,SETTINGS,COM1);
ifstream inf;
inf.open("fulldup.CPP");
start=clock();
    do
    {
    a=inf.get();
    bioscom(l,a,COM1);
    int l=bioscom(2,0,C0M1);
    s=l&0x00ff;
    cout<<s;
       if (l=127&&inf.eof())
       break;
    }
    while (!kbhit());

end=clock();
cout<<"Time requird=" << (end-start)/CLK_TCK << "sec";
getch();
}
```

Example 3A

```
 /* PROGRAM FOR BINARY FILE TRANSFER (TRANSMITTER) */

#include<dos.h>
#include<iostream.h>
#include<process.h>
#include<string.h>
#include<stdio.h>
```

```
#include<conio.h>
#include<fstream.h>
#include<io.h>
#include<fcntl.h>
#define COM 1 1
#define SETTINGS ( 0x80 | 0x02 | 0x00 | 0x00)

void main()
{
char a;
long int s;
ifstream file;
clrscr();
file.open(“C:\tcpp\xyz1.cpp”,ios::in|ios::out|ios::binary);
bioscom (0,SETTINGS,COM1);
int handel;
char buf[11]=“0123456789”;
handel=open(“C:\tccp\xyz1.cpp”,O_CREAT);
write(handel,buf,strlen(buf));
s=filelength(handel);
cout<<s;
    while(s>=0)
    {
    a=file.get();
    bioscom(1,a,COM1);
    s—;
    }
a=‘A’;
cout<<a;
bioscom(1,a,COM1);
getch();
}
```

Example 3B

/* PROGRAM FOR BINARY FILE TRANSFER (RECEIVER) */

```
#include<iostream.h>
#include<process.h>
#include<string.h>
#include<stdio.h>
#include<conio.h>
#include<bios.h>
#include<fstream.h>
#define COM1 1
#define SETTINGS ( 0x80 | 0x02 | 0x00 | 0x00)

void main()
{
long int count=0;
clrscr();
bioscom(0,SETTINGS,COM1);
ifstream infile;
infile.open("C:\TC\adcetet.cpp",ios::binary | ios::out);
    while(1)
    {
    char b=bioscom(2,0,COM1);
    long int c=b&0x00ff;
          if (c!=96)
          {
          char d=c;
          cout<<d;
          infile>>d;
          count++;
                 if (d=='*')
                 goto end;
       }
    }

end: cout<<"No. of bytes received is" << count;
infile.close();
getch();
}
```

Example 4A

/* PROGRAME FOR GO BACK TO N PROTOCOL (SENDER) */

```
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
#include<bios.h>
#include<math.h>
#include<ctype.h>
#include<dos.h>
#define COM2 1
#define SETTINGS ( 0x80 | 0x02 | 0x00 | 0x00)

int a[7],temp1,b[7],c[11],i,m=0,j,s,r;
char d;
void convert_to_binar();
void bit_check_insert();
void show_creat_stuffed_msg();
void insert_parity_bit();
void send_data();

void main()
{
char a;
    do
    {
    clrscr();
    cout << "\n"<< "Enter the character to be transfered->";
    cin >> d;
    convert_to_binary();
    bit_check_insert();
    show_creat_stuffed_msg();
    insert_parity_bit();
    cout<<endl<<"\nThe final code after calculating the check bit is =";
       for (i=0;i<11;i++)
```

```
            {
            cout << c[i]<< " ";
            }
     int q;
     bioscom(0,SETTINGS,COM2);
     retransmit:
            for(i=0;i<11;i++)
            {
            q=c[i];
            cout << q;
            bioscom(1,(char)q,COM2);
            delay(180);
            }
     cout<<"\n"<<"Data send successfully";
     cout<<"\n"<<"System is waiting for reply?";
     int wait,timer=0;

     do
     {
     wait=bioscom(2,0,COM2);
     cout<<endl<<"The ack received"<<timer<<"is";
     timer++;
            if (wait==1)
            {
            cout <<endl<<"The msg transtered by you is not received";
            goto retransmit;
            }
            if (wait==0)
            {
            cout <<endl<<"Your transfered msg is received";
            goto end;
            }
     }while(timer<=10);
  end:
  cout<<endl<<"Do you want to continue (Y/N) > ";
```

```
a=getch();
}while (toupper(a)==‘y’);
getch();
}

void conver_to_binary()
{
m=0;
temp1=int(d);
i=0;
    while(temp1>0 || temp1)
    {
    m++;
    b[i]=temp1%2;
    temp1=temp1/2;
    i++;
    }
j=0;
    for (i=6,j=0;i>=0;i—)
    {
    a[j]=b[i];
    j++;
    }
cout<<endl<<“Binary pattern of ”<<d;
    for(j=0;j<7;j++)
    cout<< a[j];
    cout <<“no of bits”;
}

void bit_check_insert()
{
r=4;
    while(i<r)
    {
    c[pow(2,i)-1]=8;
    i++;
    }
i=0;j=0;
}

void show_creat_stuffed_msg()
{
    while(i<m+r)
    {
        if (c[i]==8)
        {
        i++;
        continue;
        }
        else
        {
        c[i]=a[j];
        i++; j++;
        }
  }
cout<<endl<<"Code after inserting code bit(b is check bit):";
  for(i=0;i<11;i++)
    {
    cout<<c[i];
    }
}

void insert_parity_bit()
{
int temp1,temp2,temp3,temp4,temp8;
temp1=c[2]+c[4]+c[6]+c[8]+c[10];
temp2=c[2]+c[5]+c[6]+c[9]+c[10];
temp4=c[4]+c[5]+c[6];
temp8=c[8]+c[9]+c[10];
temp1=temp1%2;
```

```
temp2=temp2%2;
temp4=temp4%2;
temp8=temp8%2;
c[0]=temp1;
c[1]=temp2;
c[3]=temp4;
c[7]=temp8;
}
```

Example 4B

/* PROGRAME FOR GO BACK TO N PROTOCOL (RECEIVER) */

```
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
#include<bios.h>
#include<math.h>
#include<ctype.h>
#include<dos.h>
#define COM2 1
#define SETTINGS ( 0x80 | 0x02 | 0x00 | 0x00)

void send_signal(int s);
void main()
{
clrscr();
long int i, ascii=0;
char ans;
int q,c[12],a[8],pos;
bioscom(0,SETTINGS,1);
    do
    {
    rr:
    clrscr();
    i=0;
```

```
 cout <<"Received message is: ";
      do
      {
      q=bioscom(2,0,1);
      q=q&0x00ff;
             if (q==1 || q==0)
             {
             c[i] = q;
             cout <<c[i];
             i++;
             }
      }while(i!=11);
int temp1,temp2,temp4,temp8;
temp1=c[2]+c[4]+c[6]+c[8]+c[10];
temp2=c[2]+c[5]+c[6]+c[9]+c[10];
temp4=c[4]+c[5]+c[6]+c[11];
temp8=c[8]+c[9]+c[10]+c[11];
temp1=temp1%2;
temp2=temp2%2;
temp4=temp4%2;
temp8=temp8%2;
int b1,b2,b3,b4;
      if(c[0]==temp1)
      b1=0;
      else
      b1=1;
      if(c[1]==temp2)
      b2=0;
      else
      b2=1;
      if(c[3]==temp4)
      b3=0;
      else
      b3=1;
      if(c[7]==temp8)
```

```
        b4=0;
        else
        b4=1;

        int pos=((b1*1)+(b2*2)+(b3*4)+(b4*8));
        cout<<“\nvalue of”;
        cout<<b4<<“ “<<b3<< ” “<<b2<<” “<<b1<<” ”;
        cout << “Error at position “<<pos;
        cout<<“Message received:”;
        int j=0;
        for (i=0;i<11;i++)
        {
            if(i==0 || i==1 || i==3 || i==7)
            continue;
            else
            {
            a[j]=c[i];
            cout<<a[j];
            j++;
            }
        }
    ascii = ((a[0]*64)+(a[2]*16)+(a[3]*8)+a([4]*4)+(a[5]*2)+(a[6]*1));
    cout<<“\ninleger value is:”<<ascii;
    int signal;
            if (pos>=1)
            {
            signal=1;
            cout<<“\nMessage incorrect”;
            send_signal(signal);
            goto rr;
            }
            else
            {
            signal=0;
            cout<<“\nMessage correct”;
```

```
                        send_signal(signal);
                        }
    cout<<"\nSignal send is"<<signal;
    cout<<"\nReceived character is:"<<(char)ascii;
    cout<<"\nDo u want to continued (Y/N):";
    ans=getche();
    }while(toupper(ans)== 'Y');
    getch();
}
void send_signal(int s)
{
int status=bioscom(3,0,1);
    do
    {
    bioscom(1,s,1);
    cout <<"\nSending signal";
    }while(status&0*0100);
}
```

Example 5

/* PROGRAM FOR SHORTEST PATH BETWEEN NODES*/

```
#include<iostream.h>
#include<conio.h>
#include<process.h>
#include<stdlib.h>
#define permanent 1
#define tentative 0
#define infinity 10000

int i,n,j,k,min,dist[10][10],path[10];

struct state
{
int predecessor;
int length;
int label;
}state[10];

struct state *p;

void get_dis()
{
cout<<“Enter How Many Nodes : ”;
cin>>n;
    for(i=0;i<n;i++)
    dist[i][i]=0;

    for(i=0;i<n;i++)
    {
              for (j=i+1;j<n;j++)
              {
              cout<<“Enter Distance Between Nodes ”<<i+1<<” & “<<j+1<< ”:”;
              cin>>dist[i][j];
              dist[j][i]=dist[i][j];
              }
    }
}
void shortest_path(int s,int t)
{
    for(p=&state[0];p<&state[n];p++)
    {
    p->predecessor=-1;
    p->length=infinity;
    p->label=tentative;
    }
state[t].length=0;
state[t].label=permanent;
k=t;
    do
    {
              for(i=0;i<n;i++)
              {
                    if(dist[k][i]!=0 && state[i].label==tentative)
                    {
                          if(state[k].length+dist[k][i]<state[i].length)
                          {
                           state[i].pre1decessor=k;
                          state[i]. length=state[k].length+dist[k][i];
                          }
                    }
              }
              k=0;
              min=infinity;
              for(i=0;i<n;i++)
              {
                   if((state[i].label=tentative) && (state[i].length<min))
                   {
                   min=state[i].length;
                   k=i;
                   }
              }
              state[k].label=1;
    }while(k!=s);
i=0;
k=s;
    do
    {
     path[i]=k;
     cout<<path[i]+1<<" ";
      k=state[k].predecessor;
     }while(k>=0);
}
void main()
{
int  source,dest;
```

```
clrscr();
get_dis();
cout<<“\nEnter the source node number: ->”;
cin>>source;
cout<<“\nEnter the destination node number:->”;
cin>>dest;
shortest_path(source-1,dest-1);
getch();
}
```

Example 6

```
/* PROGRAM FOR TOKEN BUCKET ALGORITM (SENDER) */

#include<dos.h>
#include<iostream.h>
#include<string.h>
#include<stdio.h>
#include<conio.h>
#include<bios.h>
#include<math.h>
#include<ctype.h>
#define COM2 1
#define SETTINGS ( 0x80 | 0x02 | 0x00 | 0x00)

void main()
{
int len;
char str[5];
clrscr();
bioscom(0,SETTINGS,COM2);
    while(1)
    {
    cout << “Enter string to transfer: ”;
    gets(str);
    len=strlen(str);
           for(int i=0;i<len;i++)
```

```
              {
              cout << str[i];
              bioscom(1,str[i],COM2);
              }
       }
getch();
}
```

Example 6B

/* PROGRAM FOR TOKEN BUCKET ALGORITM (RECEIVER) */

```
#include<dos.h>
#include<iostream.h>
#include<conio.h>
#include<bios.h>
#include<time.h>
#include<fstream.h>
#define COM2 1
#define SETTINGS ( 0x80 | 0x02 | 0x00 | 0x00)

void main()
{
char a,s[5];
int i,z=0,k;
clock_t start,end;
clrscr();
bioscom(0,SETTINGS,COM2);
ll: start=clock();
k=0;
   while(s[k-1]!= '\0')
   {
   char l=bioscom(2,0,COM2);
   char n=l&0x00ff;
              if(n=='\0')
              {
              s[k]=n;
```

```
            k++;
            }

            if ((n>=‘A’ && n<=‘Z’) || (n>=‘a’ && n<=‘z’) || (n>=‘0’ && n<=‘9’))
            {
            s[k]=n;
            k++;
            }
    }
end=clock();
z=(int((end-start)/CLK_TCK))+z;
i=0;
    while(i<z && s[i]!=‘\0’)
    {
    cout << s[i];
    i++;
    }

cout <<“\n”;

    if (k<z)
    {
    cout <<“\n Still” << (z-k) << “Tocken remains ”;
    z=z-k;
    }
    else
    z=0;

    for (i=0;i<=k;i++)
    {
    s[i]=0;
    }
goto ll;
getch();
}
```

APPENDIX B

E ETHERNET STANDARDS

List of some IEEE standards.

IEEE 802.1: Standards related to network management

802.1D (1993) MAC Layer Bridges (ISO 10038)

802.1p Quality of Service and Multicast support

802.1Q VLAN processing

802.1G Remote Bridges

IEEE 802.2: General standard for the data link layer in the OSI Reference Model. The IEEE divides this layer into two sublayers — the *logical link control (LLC) layer* and the *media access control (MAC) layer*. The MAC layer varies for different network types and is defined by standards IEEE 802.3 through IEEE 802.5.

IEEE 802.3: Defines the MAC layer for bus networks that use CSMA/CD. This is the basis of the Ethernet standard.

List of most useful supplements to the IEEE 802.3 Standards:

802.3 (1985) Base standard (10B5)

802.3a (1992) 10B2 Ethernet over thin coaxial cable

802.3b (1985) Broadband Ethernet (using coaxial TV cable, now seldom used)

802.3c (1985) Improved definition of a Repeater

802.3d (1987) Definition of Ethernet for Fiber (10BFOIRL) (now seldom used)

802.3e (1987) 1Base5 or Star LAN (now seldom used)

802.3h (1991) Layer Management

802.3i (1990) 10BaseT, Ethernet over CAT-5 Unshielded Twisted Pair (UTP)

802.3j (1993) defines Ethernet over Fiber (10BF)

802.3p/q (1993) Definition of managed objects

802.3u (1995) Definition of Fast Ethernet (100BTX, 100BT4, 100BFX)

802.3x (1998) Definition of Full Duplex operation in a switched LAN

802.3y (1998) Definition of Fast Ethernet (100BT2 over low quality UTP)

802.3z Definition of Gigabit Ethernet (over Fibre)

802.3aa Definition of Gigabit Ethernet Maintainance

802.3ab Definition of Gigabit Ethernet (over UTP CAT-5)

802.3ac Definition of Ethernet VLANs

802.3ad Definition of Ethernet VLAN Trunking

802.3ae The IEEE name for its 10 Gigabit Ethernet standard, a supplement to the 802.3 standard that defines Ethernet. The 10 Gigabit Ethernet version of Ethernet operates in full-duplex mode only and supports data transfer rates of 10 gigabits per second for distances up to 300 meters on multimode fiber optic cables and up to 40 kilometers on single mode fiber optic cables. 10 Gigabit Ethernet is often abbreviated *10GbE*. The IEEE formally ratified the standard on 12 June, 2002.

IEEE 802.4: Defines the MAC layer for bus networks that use a token-passing mechanism (token bus networks).

IEEE 802.5: Defines the MAC layer for token-ring networks.

IEEE 802.6: Standard for Metropolitan Area Networks (MANs).

IEEE 802.11 Standard Applies to wireless LANs and provides 1 or 2 Mbps transmission in the 2.4 GHz band using either frequency hopping spread spectrum (FHSS) or direct sequence spread spectrum (DSSS).

List of most useful supplements to the **IEEE 802.11** Standards:

802.11a an extension to 802.11 that applies to wireless LANs and provides up to 54 Mbps in the 5 GHz band. 802.11a uses an orthogonal frequency division multiplexing encoding scheme rather than FHSS or DSSS.

802.11b (also referred to as *802.11 High Rate* or *Wi-Fi*) — an extension to 802.11 that applies to wireless LANS and provides 11 Mbps transmission (with a fallback to 5.5, 2 and 1 Mbps) in the 2.4 GHz band. 802.11b uses only DSSS. 802.11b was a 1999 ratification to the original 802.11 standard, allowing wireless functionality comparable to Ethernet.

802.11g applies to wireless LANs and provides 20+ Mbps in the 2.4 GHz band.

802.11e A wireless draft standard that defines the Quality of Service (QoS) support for LANs, and is an enhancement to the 802.11a and 802.11b wireless LAN (WLAN) specifications. 802.11e adds QoS features and multimedia support to the existing IEEE 802.11b and IEEE 802.11a wireless standards, while maintaining full backward compatibility with these standards. QoS and multimedia support are critical to wireless home networks where voice, video and audio will be delivered. Broadband service providers view QoS and multimedia-capable home networks as an essential ingredient to offering residential customers video on demand, audio on demand, voice over IP and high speed Internet access. *[Adapted from IEEE 802.11 Working Group]*

802.11n In January 2004 IEEE announced that it will develop a new standard for wide-area wireless networks. The real speed would be 100 Mbit/s (even 250 Mbit/s in PHY level), and so up to 4-5 times faster than 802.11g, and perhaps 50 times faster than 802.11b**.** As projected, 802.11n will also offer a better operating distance than current networks. The standardization progress is expected to be completed by the end of 2006. 802.11n builds upon previous 802.11 standards by adding MIMO (multiple-input multiple-output). The additional transmitter and receiver antennas allow for increased data throughput through spatial multiplexing and increased range by exploiting the spatial diversity through coding schemes like Alamouti coding.

IEEE802.16(WiMAX): Specifies WiMAX in the 10 to 66 GHz range OFDM DES3 and AES Commonly referred to as WiMAX or less commonly as WirelessMAN or the Air Interface Standard, IEEE 802.16 is a specification for fixed broadband wireless metropolitan access networks (MANs)

IEEE 802.16a (WiMAX): Added support for the 2 to 11 GHz range. OFDM DES3 and AES Commonly referred to as WiMAX or less commonly as WirelessMAN or the Air Interface Standard, IEEE 802.16 is a specification for fixed broadband wireless metropolitan access networks (MANs)

Bluetooth: Up to 2 Mbps in the 2.45 GHz band FHSS PPTP, SSL or VPN No native support for IP, so it does not support TCP/IP and wireless LAN applications well. Not originally created to support wireless LANs. Best suited for connecting PDAs,

HomeRF Up to 10 Mbps in the 2.4 GHz band FHSS Independent network IP addresses for each network. Data is sent with a 56-bit encryption algorithm.

Note: HomeRF is no longer being supported by any vendors or working groups. Intended for use in homes, not enterprises. Range is only 150 feet from base station. Relatively inexpensive to set up and maintain. Voice quality is always good because it continuously reserves a chunk of bandwidth for voice services. Responds well to interference because of frequency-hopping modulation.

HiperLAN/1 (Europe) Up to 20 Mbps in the 5 GHz band CSMA/CA Per-session encryption and individual authentication. Only in Europe. HiperLAN is totally ad-hoc, requiring no configuration and no central controller. Doesn't provide real isochronous services. Relatively expensive to operate and maintain. No guarantee of bandwidth.

HiperLAN/2 (Europe) Up to 54 Mbps in the 5 GHz band OFDM Strong security features with support for individual authentication and per-session encryption keys. Only in Europe. Designed to carry ATM cells, IP packets, Firewire packets (IEEE 1394) and digital voice (from cellular phones). Better quality of service than HiperLAN/1 and guarantees bandwidth.

OpenAir Pre-802.11 protocol, using Frequency Hopping and 0.8 and 1.6 Mb/s bit rate CSMA/CA with MAC retransmissions OpenAir doesn't implement any encryption at the MAC layer, but generates Network ID based on a password (Security ID) OpenAir is the proprietary protocol from Proxim. All OpenAir products are based on Proxim's module.

APPENDIX C

BRIEF HISTORY OF NETWORKING

The following is a brief history of computers, networking and telecommunication milestones:

1. CRT (Cathode Ray Tube) credited to Braun in 1897
2. Teletype (telegraph 5 bit) during WW1
3. ARQ (Automatic Repeat reQuest) credited to Van Duuren during WWII
    - Error checking and auto request for retransmission
4. ENIAC credited to DOD / MIT during WWII
    - Electronic Numerical Integrator And Calculator
    - Used for decoding enemy messages
    - 1st generation computer: used vacuum tubes
    - Programmed with jumpers and switches
    - MTBF (Mean Time Between Failure): 7 minutes
    - 337 multiplications per second
5. SAGE (Semi-Automatic Ground Environment) MIT 1950s
    - 23 centres for ground/air enemy detection systems
    - Error checking, keyboard & CRT terminals
    - Duplexed computers, voice grade (300-4KHz)
    - 300 baud, light pens, multiuser system
    - Magnetic core memory
    - Ground to air data Tx
    - 1st commercial use was Sabre Reservation System
6. Jacquard's Loom
    - First programmable machine
7. Transistorized Computers - 2nd Generation 1960s

- One of the 1st inventors: Cray
- Batch programming: 1 pgm @ a time
- Punch cards
- Stored programs: held in memory
- 50K instructions/second
- Ex. IBM 7905

8. CTSS (Compatible Time Sharing System) credited to Cobato/MIT in 1961
   - Time slices multiusers
9. Synchronous Orbit Communication Satellites. Idea by Arthur C. Clarke in 1945
   - Geostationary orbit around equator by Rose/Hughes Aerospace in1963
   - 36,000 miles altitude
10. LASER credited to Maiman in 1960
    - A narrow band source of optical radiation suitable for use as a carrier of info.
    - Light Amplification by Stimulated Emission of Radiation
11. T-1 Carrier System credited to Bell Labs in 1961
    - TDM (Time Domain Multiplexing)
    - 24 channels = 64 Kbps ea.
    - 1.544 Mbps (mega bits per sec)
12. RS232 developed in 1960 and revised since.
    - Standard plug and "protocol" convention between modems and machines: 25 pin
    - Europe uses V.24 compatible standard
13. Auto Equalization Techniques of Phone lines credited to Lucky et al. in 1965
    - Adapt to characteristics of telephone line to increase speed
14. Fibre Glass credited to Kao & Hockman in 1966
    - Proposed "fibre glass " optics developed at Standard Telecom Labs
15. Integrated Circuits Computers - 3rd Generation - 1967
    - SSI/MSI (Small Scale Integration/Medium Scale Integration)
    - 10 transistors/chip and 100 transistors/chip
    - Multi-user systems
    - Multitasking
16. Carterfone - FCC Decision in 1968
    - FCC decision allows other manufacturer's to use phone lines
    - Opens up competition among phone systems
17. Low-loss Fibre credited to Kapron in 1970

- Speeds: 45-90 Mbps developed at Corning Glass Works
- 1984: attained 405-565 Mbps in single mode
- Early 1990s: attained 1.7 Gbps

18. ARPA Network (ARPANET) developed by the DOD in the 1970s
    - Advanced Research Projects Agency of the Department of Defence - US
    - 1st use of Packet Switching, layered protocols
    - Beginning of the Internet
19. VLSI Integration - 4th Generation Computers developed by Intel in 1971
    - Very large scale integration: 20,000+ transistors/chip
    - Intel 4004 microprocessor - 4 bit
    - Grandparent of processors today
20. Layered Network Architecture
    - SNA: System Network Architecture IBM Mainframe
    - DNA: Digital Network Architecture DEC for DECNET
21. Ethernet developed by Xerox in 1974
    - Ether is the mysterious invisible fluid that transfers heat
    - Originally based on the ALOHA radio protocol
22. Videotex developed by Teletel (France) in the 1980s
    - Interactive video Minitel
23. Reference Model for Open Systems Interconnect developed by the ISO in 1983
    - Continuously evolving model for layering network protocols
24. AT&T Divestiture in 1984
    - Break-up of AT&T monopoly into Baby Bells
25. ISDN developed in 1984
    - Integrated Services Digital Network
    - Strong in Europe
    - A network evolving from a telephony integrated digital network supporting: voice, teletex, videotex, fax, slowscan video, etc..
26. Linux Version 0.01 released Sept 17, 1991.

APPENDIX D

STANDARD DIGITAL CODES

Computers process information in digital form. Characters are assigned a 7 or 8 bit code to indicate which character it is. This 7 or 8 bit code becomes a number (usually hexadecimal) that the computer can work with. The characters stored in a computer include:

Lower case letters:	a - z
Upper case letters:	A - Z
Digits:	0 - 9
Punctuation Marks:	. , ; : ! ? etc...
Unit Symbols:	# $ % & * etc...
Control Codes:	EOF, etc..

There are 2 major codes existing today: ASCII (pronounced ah-skee) and EBCDIC (pronounced eb-ce-dic).

APPENDIX E

ASCII–AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE

ASCII is the most popular code and is used by the majority of the computing world. ASCII itself is a 7 bit code which allows only 128 characters (27). Most applications follow IBM's Extended ASCII code which uses 8 bits and allows an addition 128 graphic characters for a total of 256 characters (28). We will be concentrating on 7 bit ASCII codes.

Format effectors

Format effectors control the movement of the cursor on the screen and the print head in a printer. The format effectors are:

BS	Backspace
HT	Horizontal Tab
LF	Line Feed
CR	Carriage Return
FF	Form Feed
VT	Vertical Tab

Communication Controls

Communication Controls are used in controlling data transmission over a communication network. They are used in both Asynchronous and Synchronous Transmissions. They are used in "handshaking".

STX	Start of Text
ETX	End of Text
EOT	End of Transmission
ENQ	End of Inquiry
ACK	Acknowledge
NAK	Negative Acknowledge
EXT	Interrupt
SYN	Synchronous idle
ETB	End of Block
EOF	End of File

Information Separators

Information separators are used to separate database enquiries and files:

FS File Separator (in a PC - used as cursor R, L, U, D)

GS	Group Separator
RS	Record Separator
US	Unit Separator

Additional Control Codes

Of the remaining codes used by the computer, the most important ones are:

NUL	Nothing character
BEL	Rings the bell!
DC1 - 4	Device Control 1 — 4
ESC	Escape — used for formatting printers & terminals
DEL	Delete — deletes characters under cursor

DC1 & DC2 are used in the Xon/Xoff software handshaking to control data transfer.

Displaying ASCII codes directly to the screen

You can type in the ASCII codes directly to the screen on IBM compatible computers. You press the “ALT” key and a 3 digit number on the numeric keypad. The 3 digit number is the ASCII decimal code for the character. You must use the numeric keypad, the QWERTY numbers will NOT work.

For example, the character “A” corresponds to the ASCII decimal code 65. To access the ASCII code directly, hold down the ALT key and type in 065 on the numeric keypad. On releasing the ALT key, the letter A will appear on the screen.

Table given below shows the ASCII codes according to decimal numbers and hexadecimal numbers. If a network sniffer or analyzer is used, it will show raw data in decimal or hexadecimal formats. You may have to perform a manual translation using given table.

Dec	*Hex*	*Name*	*Dec*	*Hex*	*Name*	*Dec*	*Hex*	*Name*	*Dec*	*Hex*	*Name*
0	0	NUL	32	20	Space	64	40	@	96	60	‘
1	1	SOH	33	21	!	65	41	A	97	61	a
2	2	STX	34	22	“	66	42	B	98	62	b
3	3	ETX	35	23	#	67	43	C	99	63	c
4	4	EOT	36	24	$	68	44	D	100	64	d
5	5	ENQ	37	25	%	69	45	E	101	65	e
6	6	ACK	38	26	&	70	46	F	102	66	f
7	7	BEL	39	27	¢	71	47	G	103	67	g
8	8	BS	40	28	(	72	48	H	104	68	h
9	9	HT	41	29	)	73	49	I	105	69	i
10	A	LF	42	2A	*	74	4A	J	106	6A	j
11	B	VT	43	2B	+	75	4B	K	107	6B	k
12	C	FF	44	2C	,	76	4C	L	108	6C	l
13	D	CR	45	2D	–	77	4D	M	109	6D	m
14	E	S0	46	2E	.	78	4E	N	110	6E	n
15	F	S1	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	|
29	1D	GS	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	DEL

ASCII code

APPENDIX F

EBCDIC–EXTENDED BINARY CODED DECIMAL INTERCHANGE CODE

EBCDIC is used mainly by IBM mainframes and compatibles. It is not common in the PC LAN world unless you are connecting to the IBM mainframe world. In order to connect, you would require either an IBM 3270 terminal emulation program or a device called a gateway.

Table 18-1 shows the EBCDIC translation table. Computers speak in binary code which is 1s and 0s. The computers do not know what the letter “A” is. Instead they speak of the letter “A” as the binary number 1100 0001. It is not easy for humans to remember binary numbers such as 1100 0001 but it is easier to remember the hexadecimal number C1. The hexadecimal number C1 is equal to the binary number 1100 0001.

The hexadecimal number C1 is equal to the decimal number 193. The table 18-1 shows both the decimal (dec) number and the hexadecimal (hex) number for the capital letter “A”. Lower case “a” is represented by the EBCDIC decimal code 129 or hexadecimal code 81.

Besides character codes such as the previous letter “A”, the EBCDIC code also defines control characters. These are characters that have special meaning. For example, the control character FF stands for Form Feed and is used by printers to advance one page or to eject a page. The decimal code for FF is 12 and the hexadecimal code is C.

Both hexadecimal and decimal codes are indicated because many times, a program or interface will report the EBCDIC code in one or the other formats. You may have to use Table given here to translate from the numerical code to the actual character.

Note: Some EBCDIC codes are not defined and have no name.

Table F.1 EBCDIC Code

Dec	*Hex*	*Name*	*Dec*	*Hex*	*Name*	*Dec*	*Hex*	*Name*	*Dec*	*Hex*	*Name*
0	0	NUL	32	20	DS	64	40	RSP	96	60	-
1	1	SOH	33	21	SOS	65	41		97	61	/
2	2	STX	34	22	FS	66	42		98	62	
3	3	ETX	35	23	WUS	67	43		99	63	
4	4	SEL	36	24	BYP	68	44		100	64	
5	5	HT	37	25	LF	69	45		101	65	
6	6	RNL	38	26	ETB	70	46		102	66	
7	7	DEL	39	27	ESC	71	47		103	67	
8	8	GE	40	28	SA	72	48		104	68	
9	9	SPS	41	29	SFE	73	49		105	69	
10	A	RPT	42	2A	SM	74	4A	¢	106	6A	|
11	B	VT	43	2B	CSP	75	4B	.	107	6B	,
12	C	FF	44	2C	MFA	76	4C	<	108	6C	%
13	D	CR	45	2D	ENQ	77	4D	(	109	6D	-
14	E	SO	46	2E	ACK	78	4E	+	110	6E	>
15	F	SI	47	2F	BEL	79	4F	ê	111	6F	?
16	10	DLE	48	30		80	50	&	112	70	
17	11	DC1	49	31		81	51		113	71	
18	12	DC2	50	32	SYN	82	52		114	72	
19	13	DC3	51	33	IR	83	53		115	73	
20	14	RES	52	34	PP	84	54		116	74	
21	15	NL	53	35	TRN	85	55		117	75	
22	16	BS	54	36	NBS	86	56		118	76	
23	17	POC	55	37	EOT	87	57		119	77	
24	18	CAN	56	38	SBS	88	58		120	78	
25	19	EM	57	39	IT	89	59		121	79	’
26	1A	UBS	58	3A	RFF	90	5A	!	122	7A	:
27	1B	CU1	59	3B	CU3	91	5B	$	123	7B	#
28	1C	IFS	60	3C	NAK	92	5C	*	124	7C	@
29	1D	IGS	61	3D		93	5D	)	125	7D	’
30	1E	IRS	62	3E	SUB	94	5E	;	126	7E	=
31	1F	IUS	63	3F	SP	95	5F	ù	127	7F	”

Dec	*Hex*	*Name*	*Dec*	*Hex*	*Name*	*Dec*	*Hex*	*Name*	*Dec*	*Hex*	*Name*
128	80		160	A0		192	C0	{	224	E0	\
129	81	a	161	A1	~	193	C1	A	225	E1	NSP
130	82	b	162	A2	s	194	C2	B	226	E2	S
131	83	c	163	A3	t	195	C3	C	227	E3	T
132	84	d	164	A4	u	196	C4	D	228	E4	U
133	85	e	165	A5	v	197	C5	E	229	E5	V
134	86	f	166	A6	w	198	C6	F	230	E6	W
135	87	g	167	A7	x	199	C7	G	231	E7	X
136	88	h	168	A8	y	200	C8	H	232	E8	Y
137	89	i	169	A9	z	201	C9	I	233	E9	Z
138	8A		170	AA		202	CA	SHY	234	EA	
139	8B		171	AB		203	CB		235	EB	
140	8C		172	AC		204	CC		236	EC	
141	8D		173	AD		205	CD		237	ED	
142	8E		174	AE		206	CE		238	EE	
143	8F		175	AF		207	CF		239	EF	
144	90		176	B0		208	D0	}	240	F0	0
145	91	j	177	B1		209	D1	J	241	F1	1
146	92	k	178	B2		210	D2	K	242	F2	2
147	93	l	179	B3		211	D3	L	243	F3	3
148	94	m	180	B4		212	D4	M	244	F4	4
149	95	n	181	B5		213	D5	N	245	F5	5
150	96	o	182	B6		214	D6	O	246	F6	6
151	97	p	183	B7		215	D7	P	247	F7	7
152	98	q	184	B8		216	D8	Q	248	F8	8
153	99	r	185	B9		217	D9	R	249	F9	9
154	9A		186	BA		218	DA		250	FA	
155	9B		187	BB		219	DB		251	FB	
156	9C		188	BC		220	DC		252	FC	
157	9D		189	BD		221	DD		253	FD	
158	9E		190	BE		222	DE		254	FE	
159	9F		191	BF		223	DF		255	FF	EO

APPENDIX G

NETWORK AND INTERNET ACRONYMS, AND URL INFORMATION

3WC	World Wide Web Consortium. Collection of professional Internet authorities that define the web.
AA	Auto Answer (modem terminology)
AAPC	Advanced Program to Program Communication
ABEND	ABnormal END of program (mainframe and programming terminology)
ACF	Advanced Communication Functions
ACIS	Automatic Customer / Caller Identification System
ACK	Affirmative aCKnowledgement (modem talk)
ACL	Access Control List
ACLS	Access Control LiSts (NT security system)
ACM	Association of Computer Machinery
ACM	Acoustical Coupling Machine
ACR	Access Control Rights
ACS	Automatic Call Sequencer.
ACU	Automatic Calling Unit (modem talk)
ADH	Average Delay to Handle (average time until connected to an agent, server, POP, or ISP)
ADMD	ADministrative Management Domain
ADS	Active Directory partition Server
ADSL	Asymmetrical Digital Subscriber Line (ISDN type connection to the net)

ADSP	Apple talk Data Stream Protocol
ADU	Automatic Dialing Unit
AFP	Apple Talk File Protocol
AGC	ARChived file: Older compression file used by PKZIP
ALD	Analog Line Driver
ANI	Automatic Number Identification
AOL	America On Line (Internet Service Provider)
ARL	Access Rights Lists (privileges & access rights to network drives, directories, or files)
ARP	Address Resolution Protocol (Novell)
ARPANET	Advanced Research Projects Agency Network (beginning of the internet)
ARQ	Automatic Repeat Request
ARU	Automatic Response Unit
ASA	Average Speed of Answer (modem talk for connection time)
ASP	Application Service Provider
ASR	Automatic Send Receive unit (modem talk)
ASU	Adaptive Start Up (modem talk)
AT	1. Asynchronous Transmission
AT	2. ATtention (modem talk)
ATT	Average Talk Time
ATD	ATtention - Dial (modem talk)
ATDT	ATtention Dial Tone (modem talk)
ATE	Asynchronous Terminal Emulation: (service that runs on Banyan)
ATH	ATtention - Hang up (modem talk)
ATM	Asynchronous Transfer Mode
ATP	Apple Talk Protocol
ATTT	ATtention Touch Tone (modem talk)
AUI	Attachment Unit Interface (IEEE spec. for LAN's interface between MAU and DTE)
AUP	Acceptable Use Policy
AV	Authentication Verification

B2B	Business To Business (also seen as b2b)
B2C	Business To Consumer
BACP	Bandwidth Allocation Control Protocol

BBS	Bulletin Board System
BCC	Block Check Character
BDC	1.Backup Domain Controller (Type of NT Server on a Microsoft Network)
BDC	2.Binary Coded Decimals
BDS	Backup Domain Server
BER	Bit Error Rate
BERT	Bit Error Rate Testing
BFS	Banyan File System
BGP	Border Gateway Protocols
BIND	Berkeley Internet Name Domain (How UNIX servers define DNS - addresses); verb: how hardware interacts with network cards / devices
BITNET	Because It's Time NETwork (academic network)
BIU	Bus Interface Unit
BN	Backbone Network
BNC	British National Connector (terminator for coaxial + cable)
BNS	Backbone Network Service
BOOTP	BOOTstrap Protocol
BSC	Binary Synchronous Communication
BSD	Berkley Software Distribution (Berkley UNIX reference)
BTS	Bit Test and Set
BYOA	Bring Your Own Access (Internet access via telnet & your own ISP)
CASE	Computer Aided Software Engineering (mainframe software utilities)
CAT	Continuous Asynchronous Transmission
CAT5	Category 5 (refers to thickness and quality of wiring)
CATV	Community Antenna TeleVision
CAU	Control Access Unit
CBX	Computer Branch eXchange
CC	Carrier Connect (modem talk)
CCIRN	Coordinating Committee for Intercontinental Research Networks
CCSA	Common Control Switching Arrangement
CCTP	Client to Client Protocol
CDA	Communications Decency Act (congressional censoring of internet)
CERT	Computer Emergency Response Team (investigate internet incidents)

CFML	Cold Fusion Markup Language
CGI	Common Gateway Interface
CGMP	Cisco Group Management Protocol (used in routers, etc.)
CHAP	Chalenge-Handshake Authentication Protocol
CHAT	Conversational Hypertext Access Technology ("talking" in real time over the internet)
CICS	Customer Information Control System
CIDR	Classless Inter-Domain Routing. Routers that contain list of resources (other routers, DNS, and services) which contain IP address
CIFS	Library files used by exe/device drivers in Windows 95 (dll format)
CIFS	Common Internet File System
CIM	Computer Integrated Manufacturing
CIS	Client Information System
CIS	Compuserver Information System
CIU	Communication Interface Unit
CIX	Commercial Internet Exchange
CLNP	Connection Less Network Protocol (OSI protocol similar to IP)
CLTP	Connection Less Transport Protocol (OSI transport layer protocol)
CLT	Communication Line Terminal
CMC	Common Mail Calls
CMC	Common Messaging Calls
CMC	Communication Management Configuration
CMIP	Certified (Common) Management Information Protocol (OSI)
CMS	Call Management System (automated voice & modem routing system)
CMS	Content Management System: came into use with dynamic data on the web with pages such as psp, asp, etc. Referes to an interface to manage the data on a website. Sometimes refering to making global changes to an entire site by editing a template
CN	Common Name
CNA	Certified Network (NetWare) Administrator
CNE	Certified Network (NetWare) Engineer
CNT	Certified Network (NetWare) Technician
COBRA	Common Object Request Broker Architecture
CODASYL	Conference On DAta SYstem Logic
COMSAT	COMmunications SATellite

CREN	Corporation for Research and Educational Networking
CRS	Content Replication System (part of the NT Backoffice suite that replicates HTML files from one server to another)
CSE	Certified Systems Engineer
CSLIP	Compressed Serial Line Interface Protocol
CSMA	Carrier Sense Multiple Access
CSMA / CA	Carrier Sense Multiple Access / Carrier Avoidance
CSMA /CD	Carrier Sense Multiple Access / Carrier Detection
CSNET	Computer + Science NETwork (academic network)
CST	Certified Systems Technician
CSU	Channel Service Unit
CSU	Channel Service Unit
CTCP	Client To Client Protocol
CTS	Clear To Send (modem talk)
DA	Destination Address (network terminology)
DACS	Digital Access on Cross-connect System
DARPA	Defense Advanced Research Projects Agency (started internet)
DBX	Digital PBX
DCD	Direct (2. Digital) Carrier Detect (modem talk)
DCD	Data Carrier Detect
DCE	1. Data Communication Equipment 2. Distributed Computing Environment 3. Data Circuit terminating Equipment
DCOM	Distributed Component Object Model (Networked OLE)
DDD	Direct Distance Dialing
DDNS	Dynamic Domain Name System
DDoS	Distributed Denial of Services. An attack on a web site, server, or ISP that shutsdown the service. Often a worm that requests so much information from the site being attacked that the overload causes a crash
DDS	Digital Data Service (IBM)
DECnet	Digital Equipment Corporation (proprietary networking protocols)
DEK	Data Encryption Key
DES	Data Encryption Standards
DHCP	Dynamic Host Configuration Protocol

DHTML	Dynamic Hyper Text Markup Language
DIW	D Inside Wiring (networking terminology common to the physical layer)
d/l	or dl, or DL: Download
DLL	Data Link Layer (2nd level of networking topology)
DLR	DOS Lan Requestor
DNA	Digital Network Architecture
DNS	**D**omain/**N**ame **S**erver/(Service) (changes addresses:www.myname.com to 116.225.25.3)
DoS	Denial of Service. Attack on an Internet server that causes it to crash. Code Red in Feb. 2002 requested so much information from servers that it caused them to shutdown when they couldn’t keep up with the requests
DOT	Dotted decimal notation for IP address in byte (1.2.4.7)
DOV	Data Over Voice (when an line is being used for both voice and data transfer) ex. ISDN lines
DPI	Distributed Protocol Interface
DSA	Directory System Agent (x.500 directory service)
DSL	**D**igital **S**ubscriber **L**ine (fast connection to the Internet vs. using the phone line)
DSR	Data Set Ready (modem talk)
DSS	Directory and Security Server (IBM’s new directory service for file servers)
DSU	Digital Service Unit
DT	1. Dial Type (modem terms)
DT	2. Dial Tone (modem terms)
DTE	Data Terminal (2. Transfer) Equipment (a LAN workstation can be one)
DTE	Dumb Terminal Emulator
DTR	Data Terminal Ready (modem signal signaling ready to receive)
DTS	Digital Termination Architecture
DUA	Directory User Agent (x.500 software)
DUN	Dial Up Networking
DWH	Data WareHousing
ECC	Error Correction Code
EFA	Extended File Attributes (Banyan / Novell)
EGP	Exterior Gateway Protocol (routing layer)

ELAN	Emulated Local Area Network
ELF	Extremely Low Frequency
EMS	Electronic Mail / Messaging System
ENS	Enhanced Network Services (Vines Service)
EOT	End Of Transmission (file transfer marker)
ERAS	Electronic Routing and Approval System (Hughe's Aircraft)
ESP	Enhanced Service Provider
ESS	Electronic Switching System
ETR	Early Token Release
EULA	End User License Agreement.
F2H	Fiber To Home: technology to provide fiber optic to residential hook-ups. Standard agreed upon in early 2003
FAP	File Access Protocol
FARNET	Federation of American Research NETworks
FC	Frame Control (networking terminology)
FC-AL	Fiber Channel Arbitrated Loop (bus technology - better/faster than SCSI)
FCS	Frame Check Sequence (Token Ring)
FD / FDX	Full DupleX (modem terminology)
FDDI	Fiber Distributed Data Interchange
FDM	Frequency-Division Multiplexing
FEC	Forward Error Correction
FIPS	Federal Information Processing Standard
FNC	Federal Networking Council (governments attempt to control the internet)
FNP	Special display effects (wipe, slide fade, ex. screen savers, and/or presentation programs)
FOSI	Formatted Output Specification Instance
FS	Frame Status (networking terminology)
FSK	Frequency Shift Keying
FSP	File Service Protocol
FTAM	File Transfer, Access, and Management (OSI service and protocol)
FTFS	Fault Tolerant File System
FTH	Fiber To Home: technology to provide fiber optic to residential hook-ups. Standard agreed upon in early 2003

FTP	File Transfer Protocol (method a file is transferred from an internet site to a local computer)
FTDP	File Transfer Protocol Daemon
GAN	Global Area Network
GENIE	General Electric Network Information Exchange
GML	Generalized Markup Language (generic HTML)
GOSIP	Government Open System Interconnection Protocol (Gvnt. agencies want to use this standard to replace TCP/IP)
GPS	Global Positioning System
GSNW	Gateway Service for NetWare
GUID	Globally Unique Identifier
HAM	Hybrid Access Method
HAP	Host Access Protocol
HDLC	High Level Data Link Control
HDX	Half DupleX
HOTT	Hot Of The Tree (electronic newsletter)
HP	Home Page (the index for a particular web site)
HSLN	High Speed Local Network
HSRP	Hot Standby Routing Protocol (to networking what FIFO is to PC's)
HTM	Unix server extension for HTML files
HTML	**H**yper **T**ext **M**arkup **L**anguage (programming for Internet web pages)
HTTP	Hyper Text Transfer Protocol (1 way information is sent over and internet; travels over the phone lines from a server to a local computer, protocols are agreed upon standards or configurations that computers use)
HTTPS	Hyper Text Transfer Protocol Secure (meaning information sent to and from is encrypted)
IAB	Internet Activities Board (oversees TCP/IP development)
IAK	Internet Access Kit (pre-set icons to AOL, CompuServer, etc.)
IANA	Internet Assigned Numbers Authority (central registry for internet assigned addresses, ports and such - ex: 225.149.149.6 (you can email: iana@isi.com))
ICAP	Internet Calendar Access Protocol (Lotus)
ICMP	Internet Control Message Protocol (IP suite - generates error message)
ICPA	Internet Consumer Protection Agency

IETF	Internet Engineering Task Force (governing body behind the internet protocols)
IGMP	Internet Group Management Protocol
IGP	Interior Gateway Protocol
IGRP	Interior Gateway Routing Protocol (Cisco proprietary protocol)
IIOP	Internet Interoperability ORB Protocol
IIS	Internet Information Server
ILS	Internet Locator Service (part of NT's Backoffice for tracking online items)
IMAP	Internet Message Access Protocol
IMAU	Intelligent Multi-Access Unit
IMS	Information Management System
InterNIC	Internet Information Center: combined providers of registration i.e., who you register your URL to the IP address with
IOS	Internet Operating System
IP	Internet Protocol
IPAD	Internet Protocol ADapter
IPS	Internet Personalization System (part of NT's Backoffice suite)
IPX	Internet Package eXchange (how PC's with different OS's transfer data)
IPX / SPX	Inter network Packet eXchange / Sequential Packet eXchange
IQY	Internet Query File
IRC	Internet Relay Chat
ISAPI	Internet Server Application Programming Interface
ISDN	Integrated Services Digital Network (Internet connection terminology - faster than using a modem, slower than T-1 connection.) Combines voice and data on same medium (or analog and digital if you prefer)
ISM	Internet Service Manager
ISOC	Internet SOCiety (non-profit organization trying to organize, research, and advance the internet)
ISP	**I**nternet **S**ervice **P**rovider
IT	**I**nformation **T**echnology (replacing the term IS for Information Systems)
JANET	Joint Academic NETwork
JAR	Java ARchive (a Java-based archiving format)
JCL	Job Control Language (mainframe programming language)
JDBC	Java Data Base Connectivity

JFC	Java Foundation Class
JUG	Joint Users Group (group of dedicated, perhaps proprietary computers)
JVM	Java Virtual Machine (Java is a new internet programming tool)
L2TP	Level 2 Tunneling Protocol
LAN	**L**ocal **A**rea **N**etwork
LAPM	Link Access Procedure for Modems
LATA	Local Area Transport Area
LBT	Listen Before Talk
LCC	Lost Calls Cleared
LCD	Lost Calls Delayed
LDAP	Lightweight Directory Access Protocol (one way server(s) manipulate multiple disks / users / and access levels)
LDDI	Local Distributed Data Interface
LEN	Low Entry Networking
LIP	Large Internet Packet
LIPS	Lightweight Internet Person Schema
LLC	Logical Link Control (level of the OSI topology)
LOS	Line Of Sight (usually refers to satellite, or dishes that transfer communication data via radio waves)
LTP	Local Talk Protocol
LU	Logical Unit
LWT	Listen While Talk
MAC	Medium Access Unit (or Multi for token ring, similar to CIU)
MAC	Media Access Control
MAN	Metropolitan Area Network (several LAN's connected together in a similar geographical region. Hospitals and Colleges use these often to share resources)
MANIAC	Mathematical Analyzer Numerical Integrator and Computer (one of the first working Mainframe computer designed and built in the early 60's)
MAP	Manufacturing and Automation Protocol
MAU	1. Multi-Access Unit (patch panel for IBM to tie to server)
MAU	2. Media Access Unit
mb	Message Board
MHS	Message Handling Service

MI	Mode Indicate (modem talk)
MIB	Management Information Bus
MIC	Mode Indicate Common (modem talk)
MILNET	MILitary NETwork (network of various branches of the military (TCP/IP))
MIME	Multi-purpose Internet Mail Extensions (Conventions for internet e-mail)
MLM	1. Mail List Manager: used for mass marketing on the Internet (spam) 2. Multi-Level Marketing.
MMC	Microsoft Management Console (NT 5.0 management program)
MML	Mathematical Markup Language
MNET	Manager Networking
MODEM	MODulate / DEModulate
MOP	Maintenance Operation Protocols
MPR	Multi-Protocol Routing
MPTN	Multi-Protocol Transport Network
MRU	Maximum Receive Units
MSDN	MicroSoft Developers Network
MSN	MicroSoft Network
MSP	Managed Service Provider
MTA	Message Transfer Agent
MTU	Maximum Transfer Unit (how large a packet/data gram IP transfers prior to TCP returning a status packet)
MUA	Mail User Agent
MUD	Multiple User Device
MUD	2.Multi-User Domain (Dungeon) - where multiple persons play games in real time via the Internet)
MVS	Multiple Virtual Systems
NAC	Network Adapter Card
NAK	Negative AcKnowledgement
NAP	Network Access Point
NAT	Network Address Translation
NAU	Network Addressable Unit
NBS	Nation Bureau of Standards
NCC	Network Control Center
NCD	Network Computing Device

NCOS	Network Computer Operating System
NDL	Network / Native Device Language
NDS	Netware Directory Services
NCG	Network Computer Group (IBM division)
NCP	Netware Core Protocol
NDIS	Network Device Interface Specification
NDS	Network Direct Services
NEST	Novell Embedded System Technology
Net BEUI	Network BIOS Extended User Interface
NetBIOS	Network Basic Input Output System
NFS	Network File System (Sun Microsystems NOS / file system)
NGI	Next Generation Internet
NHRP	Next Hop Resolution Protocol (part of the IP suite)
NIC	Network Interface Card
NIC	Network Information Center
NIP	Network Interface Protocol
NIST	National Institute of Standards and Technology
NMM	Network Management Module
NMP	Network Management Protocol
NMS	Network Management System
NNTP	Network News Transfer Protocol (part of NT's Backoffice Suite)
NOC	Network Operations Center
NOS	Network Operating System
NSF	Network File System
NTDA	NT Disk Administration tools (NTDAT)
NTFS	New Technology File System (the FAT for Microsoft NT Servers/ wkstn)
NWS	Netware Web Server
ODI	Open Data link Interface
ODIF	Open Document Interchange Format
OH	can mean either On Hook or Off Hook (modem talk)
ODINSUP	Open Data link INterface SUPport
ON	Organization Name (Banyan Vines naming convention)
ONA	Open Network Architecture
ORACLE	On-line Inquiry and Report Generator (Unix database program)

ORG	ORGanization (part of the Banyan Naming conventions)
OSA	Open System Architecture
OSI	Open System Interconnections (Topology standards for networking)
OSPF	Open Shortest Path First (‘Proposed standard’ for IGP)
OU	Organizational Unit
PAD	Packet Assemble / Dis-assemble (modem and internet talk)
PAM	Pulse Amplitude Modulation
PAP	Password Authentication Protocol
PARC	Palo Alto Research Center (Where first Ethernet network was developed by Xerox)
PAT	Port Address Translation
PBX	Private Branch eXchange
PCM	Pulse Code Modulation (converting voice to digital using a PBX system)
PCMCIA	Personal Computer Memory Card International Association
PCN	Personal Communication Network
PDL	Page Description Languages
PDM	Pulse Duration Modulation
PDN	Public Data Network (BBS)
PDS	Premises Distribution System (AT & T cabling standards for networks)
PDU	Protocol Data Unit (OSI term for a packet)
PERL	Practical Extraction and Report Language (a UNIX server based language, used to create cgi scripts and more)
PGP	Pretty Good Privacy (refers to network security)
PHP	Personal Home Page. Was used by some free servers like homestead, or geocities for a while as a default file extension. Currently it has been adopted as an extension for use with Dynamic Databases which call information into web pages. PHP Hyper text Preprocessor
PICS	Platform for Internet Content Selection
PING	Packet INternet Groper/Gopher (sending request to another computer server to see if it answers)
PKCS	Public Key Cryptography Standard
POC	Point Of Contact
POF	Point Of Fail
POP	1. Point Of Presence (internet and WWW terminology)
POP (or POP3 for version3)	2. Post Office Protocol. The protocols used by PC’s to download their e-mail from internet mail servers once PAP has verified your login

PPC	Pay Per Click. The process of on web site paying another site for every time a person clicks on the link of the first site
PPN	Peer to Peer Networking (2 computers hooked together sharing resources - no server)
PPP	Point to Point Protocol (replacing SLIP)
PPTP	Point to Point Tunneling Protocol
PR	Page Ranking: System of rating a web site or page, popular with Google search engine as method or listing the higher ranked pages in SERPs. (Search Engine Results Page)
PROARB	PROteon interrupt ARBitrator (driver for a network card)
PROTMAN	PROTocol MANager
PSAPI	Presentation Space Application Programming Interface (protocol for accessing an IBM host)
PSN	Packet Switching Network
PSTN	Public Switched Telephone Network
PUCP	Physical Unit Control Point
PWS	Peer Web Server
QBS	Quality Based System
RADIUS	Remote Authentication Dial In User Service (1 way laptops talk to LANs)
RAS	Remote Access Server
RAT	Remote Access Trojen
RBOC	Regional Bell Operating Companies (Group of major internet service providing companies)
RD	Receiving Data (modem talk)
RFC	Request For Comments - Documents that describe internet standards
RFS	Routing Information Protocol (similar to NFS but on UNIX System V)
RI	Ring Indicator
RIP	Routing Information Protocol
RJ-##	Registered (or Remote) Jack (RJ-11 for phones, RJ-45 for UTP and other cables)
RLN	Remote Lan Node
RNA	Remote Network Access (by use of Dial Up Connection)
RPC	Remote Procedure Call
RSACi	Recreational Software Advisory Council (for the Internet)

RSVP	Resource reSerVation Protocol (Cisco router term)
RTS	Request To Send (modem talk)
Rx	Receive packets
SAM	Security Accounts Manager (database of users on NT's PDC)
SAP	Service Advertising Protocol
SAP	Service Access Point
SASO	Super Automatic Switch Over - New Windows NT technology that allows data from a crashed drive / disk or failed server to be switched over to an alternate server or drive (Similar in function to the array data striping that the Banyan / Unix server uses)
SD	1. Send(ing) Data (modem talk)
SD	2. Source Address (networking terminology)
SD	3. Super Density (a type of format for the new DVD disks that allows up to 17 Gb of data to be stored on one CD)
SDN	Software Defined Network (routing through programs instead of hardware)
SE	Search Engine
SEM	Search Engine Management, or Search Engine Marketing
SEO	Search Engine Optimization (As a position in a company, Search Engine Officer)
SERP	Search Engine Results Page
SFD	Start Frame Delimiter
SFM	Services For Macintosh (protocol to allow a Mac to talk to NT servers)
SGML	Standard Generalized Markup Language
SGMP	Simple Gateway Management Protocol (pre-SNMP)
.shtml	Server Side Includes Hyper text Markup Language file extension
SIG	Special Interest Group (specialized info over the net - sponsor IRC channels, newsgroups, and various internet sites.)
SLIP	Serial Line Internet Protocol (being replaced by PPP)
SMB	Server / System Message Block protocol
SMDS	Switched Multi-megabit Data Service (connect TR to Ethernet)
SMS	System Management Server
S/MIME	Secure Multipurpose Internet Mail Extension (standard to send & receive encrypted mail)
SMDS	Switched Multi-mega-bit Data Service (new high speed transfer)
SMTP	Simple Mail Transfer Protocol

SNA	System Network Architecture (IBM proprietary architecture)
SNEWS	Secure internet NEWs Servers
SNMP	Simple Network Management Protocol
SNR	Signal to Noise Ratio
SNS	Secondary Network Server (fore-father to the BDC)
SOCKS	SOCKet Secure server
SONET	Similar to T1 or t10 except speed is 51,840,000 + Kb per second
SPE	System Policy Editor (NT's "poledit")
SPP	Sequenced Packet Protocol
SPX	Sequenced Packet eXchange
SSCP	System Services Control Point
SSL	Secure Sockets Layer (a high-level security protocol developed by Netscape)
STDA	Street Talk Directory Assistance
STML	Simple Text Markup Language
STP	Shielded Twisted Pair
STP	Secure Transfer Protocol
TAPI	Telephony Application Programming Interface
TCP/IP	Transfer Control Protocol / Internet Protocol
TGP	Thumbnail Gallery Post
TIU	Trusted Interface Unit
TOP	Technical and Office Protocol
TOU	Terms Of Use
TP	Twisted Pair (wire)
TP	Transport Protocol (old LAN networking standard protocol)
TPA	Transient Program Area (area in RAM where running programs reside)
TR	Terminal Ready (modem talk)
TR	Token Ring
TRIB	Transmission Rate of Information Bits
TSI	Time Slot Interchange (before terminals had their own processor)
TSO	Time Sharing Option (how Mainframes allocate processor time)
TTY	TeleTYpe (how the generic print driver is named)
Tx	Transmit packets
U/L bit	Universally / locally administered bit (for addressing network info)

UCE	Unsolicted Commercial E-mail
UCM	Universal Cable Module
UDP	User Datagram Protocol (Network protocol for transferring data packets - uses IP like TCP, but doesn't error check)
UID	User IDentification (internet, intranet, and server login accesses)
ULS	Universal License System (common license for cell phones)
UNC	Universal Naming Conventions
URI	Uniform Resource Identifiers (see URL)
URL	Uniform Resource Locator: The address of a web site in text form, ex: www.somesite.com. This is translated into the IP or number such as 206.22.8.45. A "Yellow Pages" for finding a webpage or site.
USN	Update Sequence Number
UTP	Unshielded Twisted Pair
UUCP	UNIX to UNIX CoPy (Currently used to describe protocol that transfers news, e-mail, data over the internet.)
VAN	Value Added Network
VAR	Value Added Reseller
VDT	Video Display Terminal
VDU	Video Display Unit
VFS	Vines File System
VINES	Virtual NEtworking Software (Banyan NOS)
VIP	Vines Internet Protocol
VMS	Virtual Machine System
VPN	Virtual Private Network
VRC	Vertical Redundancy Check (same as parity check)
VSF	Vines System Files (A Banyan / UNIX networking and operating system)
VTAM	Virtual Transmission of Automated Messaging
VTP	Virtual Terminal Protocol
W3	Short for WWW or World Wide Web
W3C	World Wide Web Consortium. Governing body of the Internet so to speak.
WAH	Work At Home
WAIS	Wide Area Information System (Info service and search engine used by many search sites like Yahoo, Infoseek, etc. to retrieve information on the Internet.)

WAN	Wide Area Network
WATS	Wide Area Telecommunications System
WDDX	Web Distributed Data eXchange
WINS	Windows Internet Naming Service (95 and NT 4.0)
WWW	**W**orld **W**ide **W**eb
WWWC	World Wide Web Consortium (governing body that proposes standards for the Web)
XML	eXtensible Markup Language (the evolution of html)
XNS	Xerox Network System
XSL	EXtensible Scripting Language - an XML style sheet language supported by the newer web browsers Internet Explorer 5 and Netscape 5

Internet Common URL extension / addresses

.alt	Alternate Lifestype (newsgroup)
.aero	Set aside for the Airline industry.
.arpa	Used for Internal networks (Address and Routing Parameter Area)
.arts	Cultural newsgroups
.asp	Active Server Page (used with dynamic databases to create pages)
asp	Application Service Provider
.biz	Businesses
.bus	Businesses
.cfm or .cfml	Cold Fusion Mark-Up Language
.cgi	Common Gateway Interface. This is a special script that runs on the server that holds the web page. It can do many things, from password protecting an area, to formatting e-mail, to calling up specific information and posting it.
.com	1. Company, 2. Common 3. Computers for news groups
.coop	Cooperation Facilities.
.edu	Educational Facility
.faq	Frequently Asked Questions (usually index page of text files you can download that give definitions / other answers to questions)
.firm	Business or Firm (domain name)
.ftp	File Transfer Protocol
.gen	General
.grp	Usually newsgroups

.gov	Government web sites.
.htm	Hyper Text Markup (UNIX servers that only accept 3 digit extensions)
.html	Hyper Text Markup Language
.http	Hyper Text Transfer Protocol
.info	Information
.int	set aside for International Treaties
.js	javascript. Developed by Netscape, it is a standard script language embedded within a web page. It can be an external file that is called by the web page
.jsp	Java Server Page
.lib	Library
.mil	Military address (Department Of Defense)
.museum	Set aside for Museum use
.name	Newer extension for domains with user names (johndoe.name)
.net	Network
.nom	a new extension for NOrMal user sites (this was suggested, but did not go over well)
.npo	Non Profit Organization
.org	Organization
.php	1. From early Homestead and Geocities servers: Personal Home Page. 2. Current dynamic data pages: ‘Perl Hyper text Preprocessor’ (this is my own take on the acronmy, feel free to correct this) Scripting that uses database resources in its’ scripting to supply web pages depending upon the content being requested. (similar to Microsoft’s .asp scripting)
.pro	set aside for use by Professional organizations (doctors lawyers, technicians, etc.)
.pub	Publications (Similar to the faq files includes html docs you can read on-line with links to related docs)
.shtml	Server Side Includes Hypertext Markup Language
.vbs	Visual Basic Scripting. Similar to javascript, made by Microsoft
.xml	eXtensible Markup Language (for dynamic data exchange in web pages). May also be seen as xhtml (extemsible Hyper text markup language)
default.asp	Usually this is an index page for ftp sites (file downloads)
index.htm	The index page for http (web) pages, also index.html

COUNTRY CODE EXTENSIONS

A - G.

ac	–	Ascension Island
.ad	–	Andorra
.ae	–	United Arab Emirates
.af	–	Afghanistan
.ag	–	Antigua and Barbuda
.ai	–	Anguilla
.al	–	Albania
.am	–	Armenia
.an	–	Netherlands Antilles
.ao	–	Angola
.aq	–	Antarctica
.ar	–	Argentina
.as	–	American Samoa
.at	–	Austria
.au	–	Australia
.aw	–	Aruba
.az	–	Azerbaijan
.ba	–	Bosnia and Herzegovina
.bb	–	Barbados
.bd	–	Bangladesh
.be	–	Belgium
.bf	–	Burkina Faso
.bg	–	Bulgaria
.bh	–	Bahrain
.bi	–	Burundi
.bj	–	Benin
.bm	–	Bermuda
.bn	–	Brunei Darussalam
.bo	–	Bolivia
.br	–	Brazil
.bs	–	Bahamas

H - M

.hk	–	Hong Kong
.hm	–	Heard and McDonald Islands
.hn	–	Honduras
.hr	–	Croatia/Hrvatska
.ht	–	Haiti
.hu	–	Hungary
.id	–	Indonesia
.ie	–	Ireland
.il	–	Israel
.im	–	Isle of Man
.in	–	India
.io	–	British Indian Ocean Territory
.iq	–	Iraq
.ir	–	Iran (Islamic Republic of)
.is	–	Iceland
.it	–	Italy
.je	–	Jersey
.jm	–	Jamaica
.jo	–	Jordan
.jp	–	Japan
.ke	–	Kenya
.kg	–	Kyrgyzstan
.kh	–	Cambodia
.ki	–	Kiribati
.km	–	Comoros
.kn	–	Saint Kitts and Nevis
.kp	–	Korea, Democratic People's Republic
.kr	–	Korea, Republic of

N - Z

.na	–	Namibia
.nc	–	New Caledonia
.ne	–	Niger
.nf	–	Norfolk Island
.ng	–	Nigeria
.ni	–	Nicaragua
.nl	–	Netherlands
.no	–	Norway
.np	–	Nepal
.nr	–	Nauru
.nu	–	Niue
.nz	–	New Zealand
.om	–	Oman
.pa	–	Panama
.pe	–	Peru
.pf	–	French Polynesia
.pg	–	Papua New Guinea
.ph	–	Philippines
.pk	–	Pakistan
.pl	–	Poland
.pm	–	St. Pierre and Miquelon
.pn	–	Pitcairn Island
.pr	–	Puerto Rico
.ps	–	Palestinian Territories
.pt	–	Portugal
.pw	–	Palau
.py	–	Paraguay
.qa	–	Qatar
.re	–	Reunion Island
.ro	–	Romania
.ru	–	Russian Federation

.bt	– Bhutan
.bv	– Bouvet Island
.bw	– Botswana
.by	– Belarus
.bz	– Belize
.ca	– Canada
.cc	– Cocos (Keeling) Islands
.cd	– Congo, Democratic Republic of the
.cf	– Central African Republic
.cg	– Congo, Republic of
.ch	– Switzerland
.ci	– Cote d'Ivoire
.ck	– Cook Islands
.cl	– Chile
.cm	– Cameroon
.cn	– China
.co	– Colombia
.cr	– Costa Rica
.cu	– Cuba
.cv	– Cap Verde
.cx	– Christmas Island
.cy	– Cyprus
.cz	– Czech Republic
.de	– Germany
.dj	– Djibouti
.dk	– Denmark
.dm	– Dominica
.do	– Dominican Republic
.dz	– Algeria
.ec	– Ecuador
.ee	– Estonia
.eg	– Egypt
.kw	– Kuwait
.ky	– Cayman Islands
.kz	– Kazakhstan
.la	– Lao People's Democratic Republic
.lb	– Lebanon
.lc	– Saint Lucia
.li	– Liechtenstein
.lk	– Sri Lanka
.lr	– Liberia
.ls	– Lesotho
.lt	– Lithuania
.lu	– Luxembourg
.lv	– Latvia
.ly	– Libyan Arab Jamahiriya
.ma	– Morocco
.mc	– Monaco
.md	– Moldova, Republic of
.mg	– Madagascar
.mh	– Marshall Islands
.mk	– Macedonia, Former Yugoslav Republic
.ml	– Mali
.mm	– Myanmar
.mn	– Mongolia
.mo	– Macau
.mp	– Northern Mariana Islands
.mq	– Martinique
.mr	– Mauritania
.ms	– Montserrat
.mt	– Malta
.mu	– Mauritius
.mv	– Maldives
.mw	– Malawi
.rw	– Rwanda
.sa	– Saudi Arabia
.sb	– Solomon Islands
.sc	– Seychelles
.sd	– Sudan
.se	– Sweden
.sg	– Singapore
.sh	– St. Helena
.si	– Slovenia
.sj	– Svalbard and Jan Mayen Islands
.sk	– Slovak Republic
.sl	– Sierra Leone
.sm	– San Marino
.sn	– Senegal
.so	– Somalia
.sr	– Suriname
.st	– Sao Tome and Principe
.sv	– El Salvador
.sy	– Syrian Arab Republic
.sz	– Swaziland
.tc	– Turks and Caicos Islands
.td	– Chad
.tf	– French Southern Territories
.tg	– Togo
.th	– Thailand
.tj	– Tajikistan
.tk	– Tokelau
.tm	– Turkmenistan
.tn	– Tunisia
.to	– Tonga
.tp	– East Timor
.tr	– Turkey

.eh	– Western Sahara
.er	– Eritrea
.es	– Spain
.et	– Ethiopia
.fi	– Finland
.fj	– Fiji
.fk	– Falkland Islands (Malvina)
.fm	– Micronesia, Federal State of
.fo	– Faroe Islands
.fr	– France
.ga	– Gabon
.gd	– Grenada
.ge	– Georgia
.gf	– French Guiana
.gg	– Guernsey
.gh	– Ghana
.gi	– Gibraltar
.gl	– Greenland
.gm	– Gambia
.gn	– Guinea
.gp	– Guadeloupe
.gq	– Equatorial Guinea
.gr	– Greece
.gs	– South Georgia and the South Sandwich Islands
.gt	– Guatemala
.gu	– Guam
.gw	– Guinea-Bissau
.gy	– Guyana
.mx	– Mexico
.my	– Malaysia
.mz	– Mozambique
.tt	– Trinidad and Tobago
.tv	– Tuvalu
.tw	– Taiwan
.tz	– Tanzania
.ua	– Ukraine
.ug	– Uganda
.uk	– United Kingdom
.um	– US Minor Outlying Islands
.us	– United States
.uy	– Uruguay
.uz	– Uzbekistan
.va	– Holy See (City Vatican State)
.vc	– Saint Vincent and the Grenadines
.ve	– Venezuela
.vg	– Virgin Islands (British)
.vi	– Virgin Islands (USA)
.vn	– Vietnam
.vu	– Vanuatu
.wf	– Wallis and Futuna Islands
.ws	– Western Samoa
.ye	– Yemen
.yt	– Mayotte
.yu	– Yugoslavia
.za	– South Africa
.zm	– Zambia
.zw	– Zimbabwe

GLOSSARY

10Base2

Shared Ethernet distributed in a “daisy-chain” fashion from one workstation to the next using BNC tee-connectors and coaxial cable. Also known as “thinnet”. The second-oldest form of commercial Ethernet.

10Base5

Shared Ethernet distributed over a special coax then tapped via transceivers which attach to the cable. The oldest form of commercial Ethernet.

10BaseT

A variant of Ethernet connecting stations via unshielded twisted pair cable using RJ-45 connectors, running at 10 Mbps. Deployed in a “star” topology, so-called because connections originate from a common point, the networking device, which may either be a hub or repeater (shared Ethernet) or a switch (switched Ethernet).

100BaseT

A variant of Ethernet connecting stations via Category 5 unshielded twisted pair cable, also with RJ-45 connectors, but running at 100 Mbps. 10 Mbps. Deployed in a “star” topology, so-called because connections originate from a common point, the networking device, which may either be a hub or repeater (shared Ethernet) or a switch (switched Ethernet).

Asynchronous Transfer Mode (ATM)

A high speed, connection-oriented switching and multiplexing technology for transmitting information across a wide area or local area network in units called cells. ATM divides information up into fixed-length cells capable of transmitting several different types of traffic simultaneously. It is asynchronous in that information streams can be sent independently, without a common clock. ATM can be described in three planes: The user plane coordinates the interface between protocols and ATM; The management place coordinates the layers of the ATM stack; and the control plan coordinates signaling, setting up and tearing down virtual circuits.

bps

Bits per second. To convert to bytes per second, divide by 8.

Bytes per second

Bytes per second. To convert to bits per second, multiply by eight.

Backbone

The generic term for LAN or WAN connectivity between subnetworks across the enterprise. Generally a conduit for traffic between multiple networks which must operate at an order of magnitude greater speed and capacity than the networks it connects. Backbones are generally bordered by either switches which consult routers or by routers.

Bandwidth

This is the range of signal frequencies that can be carried on a communications channel. While this indicates the channels information carrying capacity, it is more commonly expressed in bits per second (bps), or mega (million) bits per second (Mbps). When one says bandwidth increases, one means that network capacity and perhaps speed has gone up.

Cable Plant

The physical infrastructure (wire, connectors, cables, etc) used to carry data communications signals between data communications equipment.

Category 5 (CAT5)

A standards-based cable consisting of twisted-pair wire, with a specific number of twists per foot to reduce electrical crosstalk and provide a specific characteristic impedance (capacitive and inductive reactance) per each foot of cable. Used as an industry standard for modern cable plant, and required for Fast Ethernet. Desirable for 10BaseT as well.

Desktop

Generally considered to mean the confluence of a work location, a desk area and a computing device, distinct from a lab, which may have many computing devices. Intended to refer to one computing device, PC or workstation, which uses one network connection.

DNS

Domain Name System, or by extension, Domain Name Services.

Ethernet

Also known as CSMA/CD, it is a networking technology which relies upon collision detection to back off from simultaneous transmission. Operating at 10 Mbps, 100 Mbps (Fast Ethernet) and 1000 Mbps (Gigabit Ethernet), it is the single most commonly deployed networking technology in the LAN, and the primary one used in UCInet LANs. Ethernet is a layer-2 technology.

Ethernet Switch

A networking device which provides switched Ethernet. (see Switched Ethernet).

FDDI

A network based on a backbone of dual counter-rotating 100 Mbps fiber optic rings. One of the rings is normally designated as the primary, the other as the secondary. This even holds true if one of the point-to-point fiber optic segments becomes disabled. The counter-rotating rings are connected to single-fiber slave rings through concentrators. Bypassing inactive stations is accomplished with fiber optic switches. FDDI allows higher utilization than Ethernet. One of several technologies used in the UCInet backbone.

File Transfer Protocol (FTP)

The Internet (TCP/IP) protocol and program are used to transfer files between hosts.

Gbps (Giga bps)

A billion bits per second.

Hub

The center of a star topology network or cabling system. The term Ethernet hub typically refers to a shared-media hub. Sometimes referred to as a repeater. Supports shared Ethernet in a “star” topology over Category 5 twisted-pair wire terminated by RJ-45 data jacks.

ISP

Internet Service Provider.

Internet

The Internet; successor to DARPA-NET. Worldwide internetwork based on the TCP/IP protocol.

Internet Protocol (IP)

Part of the TCP/IP protocol suite. The layer three protocol used in a set of protocols which support the Internet and many private networks. IP provides a connectionless datagram delivery service for transport-layer protocols such as TCP and UDP.

Kbps - (Kilo bps)

A thousand bits per second.

Local Area Network - (LAN)

A network, typically Token Ring or ATM, which connects together multiple computers, printers and other network devices in a departmental or workgroup setting. It may be connected to other LANs via a backbone, typically through a router or routing device, or connected to a through a router to a WAN connection to other networks, such as the Internet.

Mbps - (Mega bps)

A million bits per second.

MAC Address

The hardware address of a device connected to a network. In Ethernet, the Ethernet address.

Micro segmentation

The technique of splitting up shared network segments by deploying switching to reduce the size of the collision domain, reduce congestion and improve throughput.

MTA

Mail Transport Agent. Unlike an MTC, the MTA actually handles sending and receiving the email to and from the system it is based upon. Most MTAs support aliasing and forwarding, as well as either the POP or IMAP protocols for clients which do not have standard MTA support upon their computer. MTAs which support POP and/or IMAP are referred to as mailhosts, and often support other functions. At UCI, MTAs are used to accept mail from the network and the outside world, then forward the mail to specific places. Mail address to user-id@uci.edu goes through campus MTAs for translation to delivery points, and actual delivery.

MTC

Mail Transport Client. Provides a user-interface for the management of email received, or the creation of it.

Multicast

A form of broadcast where copies of a packet are delivered only to a subset of all possible destinations.

Multicasting

Directing a message or a packet to some subset of all stations on a network by the use of a special destination address.

Network Segment

A portion of a network set apart from other network sections by a bridge, router or switch. Each network segment supports a single medium access protocol and a predetermined bandwidth. The more stations are on a network segment the more divided this bandwidth is. Crowded network segments lead to a condition known as congestion, where performance declines. An electrically continuous piece of a bus based LAN. Segments can be joined together using repeaters, bridges or routers. Segments may also be split apart using the same devices. The use of switches to break up segments is known as micro segmentation. See also shared and switched Ethernet.

PH

A program used on several campuses, including UCI, as an interface to the QI database. Used as an electronic phone book.

PH/QI

Used, generally interchangeably with PH, as a reference to PH, the UCI campus electronic phone book.

QI

The database UCI uses with ph, as part of the ph/qi phone book.

Quality of Service

This is a networking term which may be used in one of two ways. In the first way, it represents a quality of networking. In the second, referred to usually as “QoS”, it represents a guarantee or commitment to not only a particular quality of network service but also a particular rate or minimum rate of data delivery, as well as maximum times between packets of data. Used where applications are sensitive to delays, such as video conferencing. Initially a feature of ATM, it is now being incorporated into the TCP/IP protocol and will eventually be available as a service on non-ATM networks. A statement that QoS is provided is distinct from one which says QoS guarantees are provided. UCInet provides a good quality of service, but not QoS guarantees as part of Basic Network Services.

ROUTING

The function of determining the route a packet should take from a subnet to get to another subnet. A component function of all internetworks, or internets. The process of delivering a message across a network or networks by the most appropriate path.

Route

The patch that network traffic takes from its source to its destination.

Router

A system responsible for making decisions about which of several paths network traffic will take, and for keeping track of routing information which is being passed along a network be one of several different possible protocols. To do this a router uses a routing protocol to gain information about the network and uses algorithms to choose the best router based on several criteria known as route metrics. In OSI terminology, a router is a Network Layer intermediate system. See also IP router.

Shared Ethernet

A network segment which has multiple nodes connected, and where available bandwidth is divided among users in a dividing effect. May be deployed using 10Base5, 10Base2 or 10BaseT (the latter being a hub). The most widely deployed technology in UCInet.

Simple Network Management Protocol (SNMP)

Both a way of obtaining and storing information about network devices. A protocol designed to manage networking devices. SNMP capable devices keep statistics on their operation, if instructed to do so, which may be remotely fetched and analyzed by central management stations.

Star

A network topology where each node is connected to a central hub.

Subnet

A portion of a network, which may be physically independent, that shares a network address with other portions of the network and is distinguished by a subnet number. Subnets are created and supported by the use of routing.

Subnet Address

The subnet portion of an IP address.

Switch

A device which logically connects to network stations through a network fabric.

Switched Ethernet

An Ethernet technology deployed from a central box over Category 5 twisted-pair wire or fiber optic, and which allows the full utilization of bandwidth for each network conversation by switching connections point-to-point between stations talking to each other, providing in effect a dedicated connection. Considered to be an order of magnitude faster than its shared counterpart.

TCP/IP

Transport Control Protocol / Internet Protocol. The protocol of the Internet and most internets and many intranets.

Transmission Control Protocol (TCP)

A layer-four protocol in the set of protocols which support the Internet and many private networks. TCP is the TCP portion of TCP/IP, and provides a guaranteed transport service.

UCInetId

(Also UCInet-id or Ucinet-Id) Unique string of characters representing an entry in the campus electronic phone book and in the campus authentication database. When coupled with the string '@uci.edu', it becomes a campus email address. UCInet has services (ph/qi, Kerberos authentication, MTAs, etc.) which consult these databases in order to authenticate users, forward or deliver email, and assist users in finding other users on the network, by phone or location.

Virtual Local Area Network (VLAN)

Individual workstations, rather than being directly connected to a shared media, are instead connected to an intelligent device such as a switch which has the capability through software to define LAN membership. This permits a systems administrator to resegment the LAN without changing the physical arrangement. It also allows, with some switching technologies, the ability to support multiple subnets on a single switch where a series of router interfaces were previously required.

Wide Area Network (WAN)

A network which covers a larger geographical area than a LAN or a MAN and where telecommunications links are implemented, normally leased from the appropriate Private Telephone Operator(s). Examples of WANs include packet switched networks, public data networks and Value Added Networks (VANs).

WAN Connection

A network connection, usually through a router or an ATM switch, which connects two geographically distanced networks together.

INDEX

(DES) Data Encryption Standard 311

.200 Series 118

10 BASE 2 92

10 BASE 5 92

10 BASE-F 94

10 BASE-FB 95

10 BASE-FL 95

10 BASE-FP 94

10 BASE-T 93

10 BROAD 36 94

100 BASE-T4 236

100 Base-X 236

11.3 TCP 298

1-Persistent CSMA 221

2-Wire modem 163

4-Wire modem 164

A

A Bank's Automatic Teller Machine 14

A Private Key 312

A proxy Agent 318

A Public Key 312

A Simplex Protocol for a Noisy Channel 193

A Simplex STOP-AND-WAIT Protocol 191

ABM (Asynchronous Balanced Mode) 207

Acknowledged connectionless service 171

Acknowledged connection-oriented service 171

Acknowledgement number 300

Active 206

Active hubs 106

ACTS (Advanced Communication Technology Satellite) 240

Adaptive 249

Adaptive algorithms 249

Additional features 209

Address resolution 285

Addressing 293

ALOHA 217, 238

Amplitude 8

Amplitude modulation 17

An unrestricted simplex protocol 190

Analog signal 7

Application layer 63, 65

Application layer Services 321

Application process 2

Architecture 115

ARCnet 68

ARM (Asynchronous Response Mode) 207

ARP (Address Resolution Protocol) 285

ARP operation 286

ARPANET 63, 73

ASN. 1 Abstract Syntax Notation 1 315

Asymmetric 292

Asynchronous modem 164

Asynchronous transmission 10
ATM 264
ATM cells 274
Atmospheric noise 30
Attenuation 150
Authentication 307, 313
Automatic repeat request 194
Autonomous system 273

B

B channel 119
Balanced asynchronous 210
Balanced link configuration 204
Bandwidth 18
Bandwidth limited signals 128
Baseband co-axial cable 134
Baseband signal 16
Basic access 119, 120
Bellman-ford method 250
Benefits 114
Bessel 24
Bessel functions 24
BIT Rate 12
Bit-Oriented framing 174
Block schematic of a modem 166
BNC 98
Boolean 188
Breaking mono-alphabetic ciphers 309
Bridge 268
Bridges/Switches 100
Broadband co-axial cable 136
Broadband ISDN 110, 121
Broadcast 38, 257
Broadcast routing 257
Burst errors 178
Bus topology 54

C

Cable topologies 95
Cabling and connector standards 91
Caesar cipher 308
Called entities 79
Carrier frequency 37
Carrier sense 217
Carrier signal 16
Carson’s rule 26
Categories of satellites 145
CCITT 83
CDMA 241
Cellular digital packet data 57
Cellular telephone 148
Cellular topology 57
Centralised network model 48
Chain block cipher (CBC) 312
Channel 5
Channel allocation 215
Channel speed 12
Channels 4
Character stuffing 174
Character-Oriented framing 173
Checksum field 69
Choke packets 263
Cipher Feedback Mode (CFM) 312
Circuit 5
Circuit assurance phase 160
Clark’s solution 302
Closed loop 258
Closed loop algorithms 259

Co-axial cable 134
Code violation-oriented framing 175
Collision assumption 216
Color division multiplexing 43
Communication 1
Communication in the internet works 279
Communication in the internet works as follows 279
Communication system 2
Communications network 3
Comparison of virtual circuit and datagram subnets 247
Computer center 45
Computer communication 1
Computer network 46
Concatenated virtual circuits 269
Concentrators 105
Congestion 64, 257
Congestion prevention policies 259
Connect 291, 292
Connection 278
Connectionless network service 277
Connection oriented communication 66
Connection oriented service 80
Connectionless 66
Connectionless internetworking 270
Connectionless services 81
Connection-oriented service 172
Connector standards 96
Cons 246
Contention 223
Continuous time 216
Control plane 68, 122
Cosmic noise 30
Countries 319
Crash recovery 297
Cryptography 308
CSMA (Carrier Sensing Multiple Access Protocols) 221
CSMA/CD 223
Cyclic Redundancy Check (CRC) 180

D

D channel 119
Damaged frame 176
Data communication 125
Data link layer 61, 170, 277
Data switching 6
Data transfer phase 160
Datagram (DG) 87
DB-25 97
DB-9 97
DCA 72
DCE 3
DCE-DCE connection 156
Descrambler 165
Design issues 243
DES-operational modes 312
Destination port 300
Deviation 22
Differential manchester 136
Digital logic 184
Digital signal 8
Dijkstra's algorithm 255
Disconnect 292
Disconnect phase 161
Discrete 9
Distance vector routing 253
Distortion 151
DNS 306

DNS (Domain Name System) 65, 318
DNS name space 319
Domain name service 65
Domain-name 320
Domains 319
Dotted decimal notation 282
downlink 146
Downward multiplexing 297
DPSK 94
DSE 6
DTE and DCE ports 156
DTE/DCE interface 155
Dual cable system 137
Dumb terminal 3
Duplex 5
Dwell time 237
Dynamic channel allocation 216
Dynamic routers 105

E

EIA-232-D digital interface 155
EIA-232-D interface specifications 157
Electrical specification 155, 158
Electronic cash registers 14
Electronic code book (ECB) 312
Electronic mail system 15
Elementary data link protocols 186
Elements of transport protocols 293
E-mail 74, 322
Encryption with public keys 312
Equipment 6
Equipment readiness Phase 160
Equivalent noise temperature 34
Erlang 219
Error control 175
Error detection 176, 178
Error correction 176
Ethernet 68
Explicit feedback 259
Exterior gateway protocol 272
External Noise 30
Extranet 90
Extra-terrestrial/Space noise 30

F

Facsimile machine 14
Fair queuing algorithm 263
Fast ethernet 235
Fast select 88
FDDI (Fiber Distributed Data Interface) 231
FDDI frame format 233
FDDI specifications 234
FDDI transmission media 232
FDM 37, 239
File transfer 75
Fire walls 275
Firewalls 275
First kind 283
Flooding 250, 257
Flow based routing 251
Flow control 62, 186
Flow control and buffering 295
Fourier analysis 125
Fourier series 125
Four-wire circuits 6
Fragment offset 280
Fragmentation 273
Fragments 274
Frame 188
Framing techniques 173

Frequency 8, 21
Frequency division multiplexing 37
Frequency modulation 21
Frequency reuse pattern 148
Frequency reuse principle 148
FTP 65, 278
Full duplex 5
Full duplex modem 162, 163
Functional grouping 121
Functional specification 155, 159
Functions of protocols 82
Fuzzball 73

G

Gateway 268
Gateway addressing 110
Gateways 109
Generic 319
GEO satellites 146
Geosynchronous 146
Gigabit testbeds 75
GPS 147
Group membership 288
Guard bands 38
Guided media 132
GWL 147

H

H channel 119
Half duplex 5
Half duplex modem 162
Hamming codes 178
Hamming distance 176
Handoff 149
HDLC (High Level Data Link Control) 203
HDLC classes of procedures 210
HDLC flow control 209
HDLC frame format 207
HDLC frame types 209
HDLC logical link configuration 204
HDLC logical states 206
HDLC transmission states 206
Header checksum 281
Heterogeneous network 62
High speed lans 231
Home page 326
Hop-by-Hop choke packets 264
Horn antenna 144
Host 70
Host-to-Network layer 64
HTML (Hyper Text Markup Language) 330
HTTP (Hyper Text Transfer Protocol) 65, 278, 327
HUB 53, 106

I

I.100 Series 118
I.200 Series 118
I.300 Series 118
I.400 Series 118
I.500 Series 118
I.600 Series 118
I.700 Series 118
IBM token ring 68
ICMP (Internet Control Message Protocol) 284
Idle 206, 223
IEEE 802.3: ethernet 227
IEEE 802.4 : token bus 229

IEEE 802.5 : token ring 230
IEEE standards 227
IGMP (Internet Group Message Protocol) 288
Implicit feedback 259
IMPs (Interface Message Processors) 70
Information frame 209
Interactive system 14
Interfaces 79
Interior gateway protocol 272
Intermodulation 40
Intermodulation noise 40
Internal noise 31
Internet control protocol 284
Internet layer 64
Internet mail protocols 323
Internet protocol (IP) 278, 279
Internet tools 322
Internetwork routing 272
Internetworking devices 268
Internetworks 50
Intranet 90
Ionosphere 143
Ionospheric propagation 143
IP (Internet Protocol) 278
IP Addresses 281
IS (Initialization State) 206
ISDN 110
ISDN Channels 119
ISDN User Access 120
Isochronous transmission 10, 12
ITS (Information Transfer State) 206

J

Jitter control 267

K

Keepalive timer 303
Key length 311
Kind 188

L

LAN 4
Layered approach 78
LDS (Logically Disconnected State) 207
Leaky bucket algorithms 260, 261, 262
LEO satellites 147
Line configuration 51
Linear (BUS) topology 95
Line-of-Sight propagation 143
Link Access Procedure, D Channel (LAPD) 87
Link Access Protocol (LAP) 87
Link Access Protocol Balanced (LAPB) 87
Link state routing 255
LISTEN 291
LLC (Logical Link Control) 227
Load shedding 266
Local bridges 101
Local-Area Network (LAN) 48
Logical Link Control (LLC) 87
Loop structure 206
Loose source routing 281
Lost frame 176

M

Managed nodes 317
Management plane 68, 122
Management stations 318
Manchester 136

Man-made/Industrial noise 31
Maximum deviation 23
MCS (Master Control Station) 240
Mechanical specification 155, 158
MEO satellite 146
Mesh topology 52
Metropolitan-Area Network (MAN) 49
MIB (Management Information Base) 318
Milk 266
Milnet 72
Mobile Switching Center (MSC) 148
Modems 3, 161
Modes of operation 207
Modulating frequency 22
Modulating voltage 22
Modulation 16, 21, 24
Modulation index 19
Modulo 2 arithmetic 180
Mono alphabetic substitution 308
MRI scans 76
Multicast routers 288
Multidestination routing 257
Multidrop channels 4
Multimode fiber 140
Multi-Mode Fiber 233
Multimode graded-index fiber 140
Multimode step-index fiber 140
Multiplexing 36, 296
Multipoint 51
Multiprotocol router 269

N

Narrowband ISDN 110
National institute of science and technology 310
Negative acknowledgement and retransmission 176
Network 278
Network architecture 58
Network components 91
Network interface card (NIC) 99
Network layer 62, 277
Network layer in the internet 278
Network management 306
Network security 306
Network topologies 52
News 75
NNTP 65
No carrier sense 217
Noise 151
Noise figure 33, 34
Non-adaptive 249
Non-adaptive algorithms 249
Non-discrete 7
Non-interactive system 15
Non-persistent CSMA 222
Nonrepudiation and integrity control 307
Non-transparent fragmentation 275
Novel netware 68
NRM (Normal Response Mode) 207
NSF 72
NSFNET 73
Nyquist's theorem 131

O

Object identifier 317
Objects 318
Octet strings 317
Offline systems 13, 14
Online 13

Online systems 13
Open loop 258
Optical FDM 43
Optical fiber 138, 233
Optimal routing 250
Options 281
OSI 66
OSI reference model 60

P

Packet 188
Packet filter 276
Packet length 69
Packet type 69
Page 326
Paging 149
Parallel transmission 9
Passive hubs 106
PDN (Public Data Network) 83
Peer entities 79
Peer processes 58
Percentage modulation 19
Permanent virtual circuit (PVC) 83, 84
Persistence timer 303
Phase 8
Physical connection 152
Physical layer 61, 277
Physical layer standards 154
Physical security 307
Piggybacked 197
Point-to-point 51
Polling 237
Polynomials 183
Port 298
Positive acknowledgement 176
Positive acknowledgement with retransmission 194
Persistent CSMA 222
PPP (Point-to-Point Protocol) 212
Presentation layer 63
Primary access 119, 120
Private keys 312
Procedural specification 155, 160
Product ciphers 309
Propagation 142
Propagation modes 139
Propagation of specific signals 143
Pros 246
Protocol 58, 281
Protocol hierarchies 57
Protocols 3, 81
PS2/MINI-DIN 99
Pure ALOHA 217

Q

Questionnaries 122

R

Radio frequency allocation 141
RARP (Reverse Address Resolution Protocol) 286
Receive 292
Record route 281
Reference points 121
Reference station 238
Remote bridges 101
Remote login 75
Repeater 96, 107, 144, 268
Residual error rate 180

Resolver 318

Retransmission after time out 176

Retransmission policy 260

Retransmission timer 303

Reuse factor 148

Reusing cells 149

Reverse Address Resolution Protocol (RARP) 286

Reverse path forwarding 257

Ring topology 55

RJ-11 98

RJ-45 98

Roaming 150

Routers 104

Routing 248

Routing algorithm 248

RSA algorithm 313

S

Satellite communication 144

Satellite networks 237

Scrambler 165

Searching for information on the internet 327

Second kind 283

Secondary hub 56

Secrecy 307

Secret key algorithms 310

Security 281, 306

Segment 299

Segmentation 274

Segmented topology 96

Send 292

Seq. and Ack 188

Seq-nr 188

Sequence number 300

Serial transmission 10

Service primitives 290

Service provider 79

Service user 79

Services 80, 115

Session layer 62, 314

Session routing 249

Shannon's 131

Shielded 133

Shielded twisted-pair (STP) cable 133

Shortest path routing 249

Shot noise 32

Signal cable system 137

Signal to noise ratio 33, 131

Signals 7

Silly window syndrome 302

Simplex 5

Single channel assumption 216

Single mode 139

Single-mode fiber 233

Sliding window protocol with go-back-N 200

Sliding window protocol with selective repeat 201

Sliding window protocols 196

Slip-serial lines IP 211

Slotted ALOHA 219

Slotted ALOHA with non-owner 220

Slotted ALOHA with owner 221

Slotted time 217

SMTP 65

SNMP (Simple Network Management Protocol) 317

SNMP agent 317

Sockets 298

Solar noise 30

Source port 300

Source routing bridges 103

Space propagation 143

Spanning tree 257

Spanning tree protocol 103

Spine topology 53, 95

Static channel allocation 215

Static routers 105

Station model 216

Statistical TDM (STATDM) 42

Stop-and-wait sliding window protocol 198

Store-and-forward message 15

Strict source routing 281

Subcarrier 38

Subnet 70

Subnet mask 284

Subnets 283

Subnetting 283

Substitution ciphers 308

Supervisory frame 209

Surface propagation 143

Switch 104

Switched virtual circuit (SVC) 83, 84

Switching hub 107

Symbols

Symmetric 292

Symmetrical link configuration 205

Synchronization 63

Synchronous modem 165

Synchronous TDM 41

Synchronous transmission 10, 11

T

TCP 64

TCP congestion control 302

TCP connection management 301

TCP header length 300

TCP Protocol 299

TCP reference models 66

TCP timer management 303

TCP transmission policy 301

TCP/IP protocol suit overview 277

TCP/IP Vs OSI model 278

TDM 239

Television 38

TELNET 65

Terminal interface processor 72

Terrestrial microwave 143

The arpanet 69

The data link layer in the internet 210

The digital pi 119

The Internet 74

The internet message access protocol (IMAP) 323

THE I-Series Recommendations 117

The link level 86

The maximum date rate of a channel 131

The multipurpose internet mail extensions (MIME) 323

The NSFNET 72

The packet level 87

The physical level 86

The Post Office Protocol (POP) 323

The Simple Mail Transfer Protocol (SMTP) 323

The TCP segment header 300

The TCP service model 298

The TCP/IP reference model 63
The user interface 112
Thermal noise 30
Thermal noise 31
Threshold 303
Throughput 218
Time division multiplexing 40, 41
Time stamp 281
Time to live 280
Timed wait 304
Time-to-live 320
TIP 72
Token 55
Token bucket algorithm 262
Token management 63
Topology 52
Transient 206
Transit-time noise 33
Transmission 9, 223
Transmission control protocol 64, 278, 298
Transmission impairment 150
Transmission media 132
Transmission structure 119
Transmitting 149
Transparent bridge 103
Transparent fragmentation 274
Transport control 69
Transport layer 243, 277, 290
Transport layer 290
Transport protocol data unit 291
Transport service primitives 290
Transposition ciphers 309
Tree topology 56, 96
Triangulation 146
Troposphere 142
Tropospheric propagation 143
Tunneling 271, 272
Twisted-pair cable 132
Two-way-alternate 5
Two-way-simultaneous 5
Two-wire 6
Type 320
Types of access protocols 217
Types of modems 162
Types of stations 204

U

UDP 65, 304, 318
UML 147
Unacknowledged connectionless service 171
Unbalanced asynchronous 210
Unbalanced link configurations 204
Unbalanced normal 210
Unguided media 141
Universal resource locators 327
Unnumbered frame 209
Unshielded 133
Unshielded Twisted-Pair (UTP) Cable 133
Uplink 146
Upper layers 306
UPS 307
Upward multiplexing 296
Urgent pointer 300
USB 99
Usenet news 324
User data protocol 304
User datagram protocol 65
User plane 68, 122

V

Vampire tap 135

Virtual Circuit (VC) 87

Virtual meeting 76

W

WAN subnet 172

Wavelength division multiple access 43, 225

Web page and hypertext markup language (HTML) 327

Web site 326

Weighted fair queuing 263, 264

Wide-Area Network (WAN) 50

Win 266

Window size 197

Wireless networks 77

WWW (World Wide Web) 75, 326

X

X.25 Protocol 83

X.25 State diagrams 88